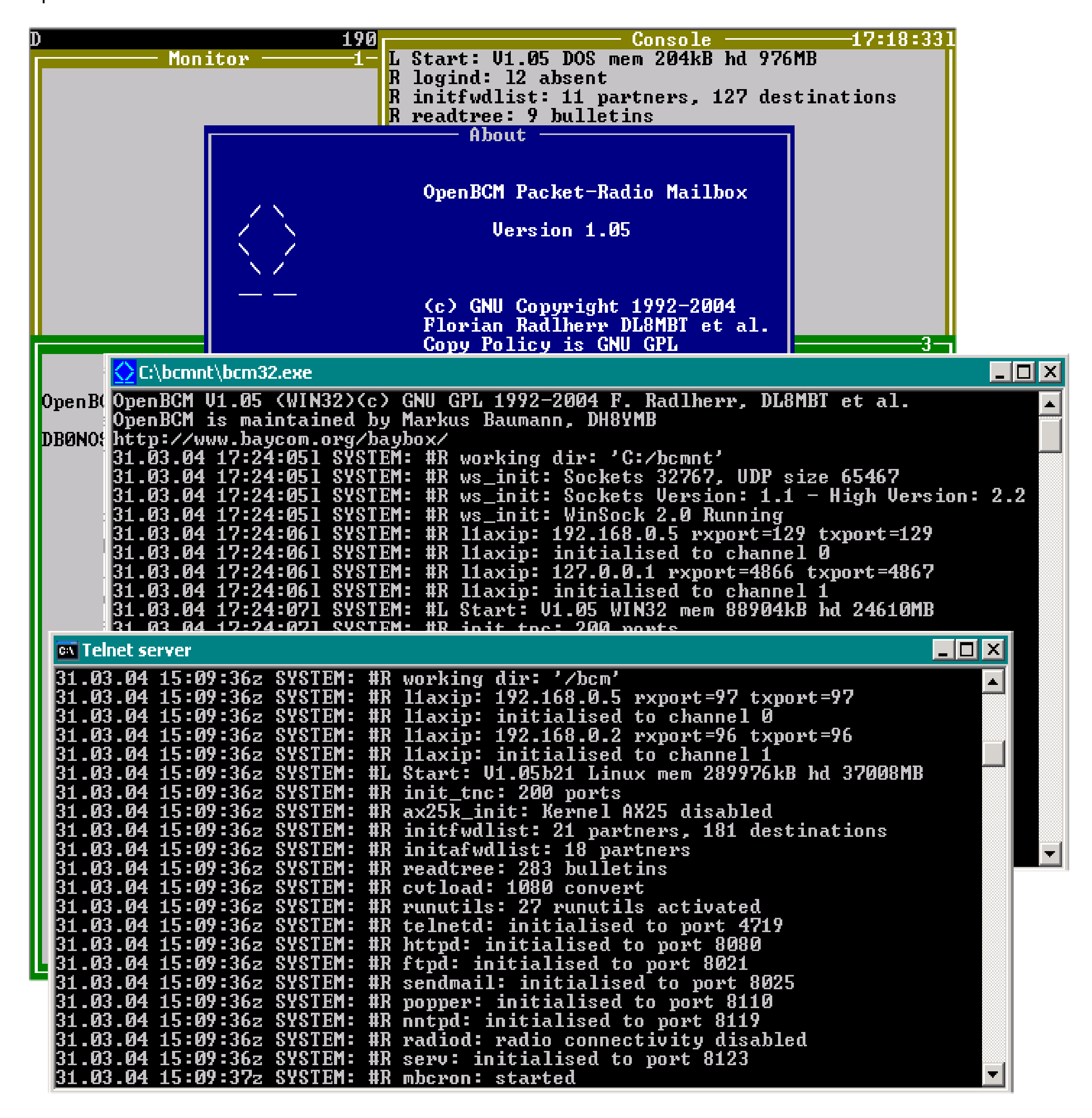
Dokumentation
zur OpenBCM-Mailbox v1.05

Die komplette Anleitung zur Installation und zum Betrieb einer
OpenBCM-Mailbox



Original-Dokumentation von Florian Radlherr DL8MBT <flori@baycom.org>,
Ergänzungen und Erweiterungen von
Dietmar Zlabinger OE3DZW <dietmar@baycom.org>,
Johann Hanne DH3MB <jonny@baycom.org> und
Markus Baumann DH8YMB <dh8ymb@web.de>

Stand: 02. April 2004

Inhaltsverzeichnis

1. Vorwort und Allgemeines

1.1. Vorwort

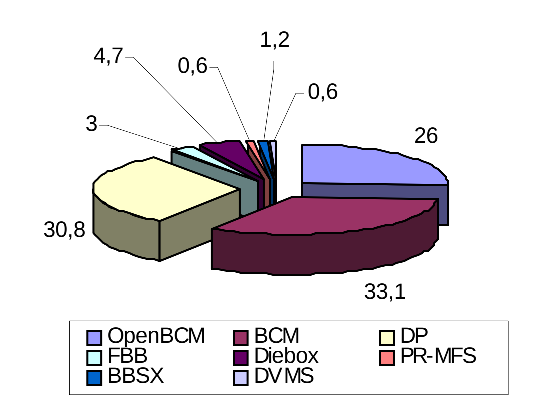
Seit dem Release von OpenBCM v1.04 im Dezember 2003 hat sich wieder einmal einiges getan. Passend zur 20. PR-Tagung in Darmstadt ist OpenBCM v1.05 erschienen. Als größte Neuerung dürfte dabei der FTP-Server ins Gewicht fallen, den Jens DH6BB in mühsamer Arbeit in den Quellcode implementiert hat (an dieser Stelle noch einmal ein großes Dankeschön)!

Gegenüber der v1.04 ist sicherlich auch die Erweiterung des LOG Befehls für den ein oder anderen interessant. Externe (Run-)Utilities zur Auswertung der Log-Dateien sind nun eigentlich überflüssig, da man jederzeit aktuelle Zusammenfassungen in Form einer Loginstatistik abrufen kann, auf Wunsch auch tage-, wochen- oder monatsweise, vierteljährlich, im Jahr usw...

Bei der Vorbereitung auf den Vortrag "Neue Features bei OpenBCM" für die PR-Tagung in Darmstadt am 03.04.2004 habe ich auch aktuelle Nutzungsdaten aus der Userdatenbank von DB0ZWI für den Amateurfunk ausgewertet. Es ergibt sich dabei folgendes Bild: in DL gibt es momentan 169 Packet-Radio-Mailboxen und genau 100 davon laufen mit BCM/OpenBCM – ein erfreuliches Ergebnis, das die hohe Akzeptanz in dies Mailboxprojekt widerspiegelt:

Mailboxanzahl und prozentualer Anteil:

44	**OpenBCM**	**26,0%**
56	**BCM**	**33,1%**
52	**DP**	**30,8%**
2	**BBSX**	**1,2%**
1	**DVMS**	**0,6%**
1	**PR-MFS**	**0,6%**
5	**FBB**	**3,0%**
8	**DieBox**	**4,7%**
169	**Boxen in DL**	



Insbesondere erfreulich ist weiterhin die Tatsache, dass sich auch zahlreiche Mailboxsysops aus dem CB-Bereich für OpenBCM interessieren und bereits einsetzen.

Besonders die Linux-Version der OpenBCM-Mailbox wird mit zunehmendem Maße und mit großem Erfolg eingesetzt (von 44 OpenBCM-Installationen laufen 40 unter Linux, 3 unter Windows und 1 unter DOS). Begünstigt wurde diese Entwicklung auch durch die zunehmende Popularität und vereinfachte Installation von Linux als Anwenderbetriebssystem.

Denjenigen Sysops, die noch eine alte Mailbox-Version einsetzen, sei nochmals dringend empfohlen, auf die aktuelle OpenBCM v1.05 umzusteigen. Was insbesondere bei einem Update von BCM v1.42n auf OpenBCM v1.05 beachtet werden sollte, ist im Kapitel "22. Umstellung von BCM v1.42n auf OpenBCM" zusammengefasst.

Es bleibt am Ende des Vorwortes wieder einmal übrig, noch denen zu danken, die durch endlose Tests und qualifizierte Berichte zur heute verfügbaren Stabilität und Funktionalität beigetragen haben, sowie all denen, die sich aktiv an Weiterentwicklung beteiligt haben!

31.03.2004 - Markus, DH8YMB

1.2. Entstehungsgeschichte

- August 1991: Erster Entwurf einer Spezifikation für ein Mailbox-Dateisystem. Diese war der heutigen Realisierung bereits sehr ähnlich.
- Januar 1992: Entwurf und Entwicklung einer universellen Benutzeroberfläche mit Editor und mit Multitasking-Kern. Bei der Entstehung dieser Teile wurde noch nicht an einen Einsatz in einem Mailboxprogramm gedacht.
- Mai 1992: Entscheidung: "Es wird eine Boxsoftware entwickelt"; Entwicklung von ersten Modulen für die Mailbox, insbesondere BID-Verwaltung, Userdaten-Verwaltung, Suchfunktionen, Datei-Semaphoren, etc.
- Juni 1992: Erstellung des Boxgerüstes, TNC-Anschluss, Kommandoauswertung, etc.
- Juli 1992: Integration der Mailboxteile mit der Benutzeroberfläche, Einsatz des Multitasking-Schedulers
- August 1992: Erstellung und Test der Store&Forward-Routinen, erste Systemtests über Funk unter dem Rufzeichen DF0AR
- September 1992: Inbetriebnahme der neuen Box bei DB0AAB. Zunächst am Standort von DL8MBT über einen Funklink. Nach fortgeschrittener Fehlerbehebung Inbetriebnahme bei DB0AAB auf der Fachhochschule München
- Oktober 1992: Inbetriebnahme der BayCom-Mailbox an den Knoten DB0LNA und DB0RGB
- November 1992: Herausgabe der Version 1.00 auf der Interradio in Hannover
- Februar 1994: Version 1.17, Mehrsprachigkeit der Mailbox
- September 1994: Binäre Nachrichten
- November 1994: Version 1.35, erste Linux-Version
- Februar 1995: Version 1.36
- Juni 1995: Vorstellung von PC/FlexNet auf der HamRadio, erste FlexNet-Version der Mailbox
- Januar 1996: Version 1.37 für FlexNet 3.3e
- Oktober 1996: Version 1.38 für Linux und Windows NT, User-S&F
- Juni 1997: Version 1.39 mit HTTP-Interface
- Juni 1998: Version 1.40 mit einer Fülle an neuen Features: 7plus-Erkennung, komprimierter S&F nach F6FBB, Mailing-Listserver, File-Server, MD2/MD5-Passwortverfahren, YAPP-Protokoll, neues Run-Utility-Interface, Unterstützung des Linux Kernel-AX.25 und einer Menge Bugfixes
- November 1999: Version 1.42 Jahr-2000-fähig, neues Protokoll für die Weiterleitung von White-Page-Informationen, Bugfixes, deutlich verbesserte Unterstützung von CB-Funk. Neue Lizenz (GPL).
- Oktober 2001: Version 1.44 zahlreiche neue Features, viele kleinere Änderungen am Quellcode, u. a. diverse Patche von DF3VI
- September 2002: Kurz nach dem Erscheinen der Version 1.46 wird der Name von "BCM" in "OpenBCM" geändert. Der Versionzähler beginnt mit dem neuen Namen nun wieder bei 1.00.
- Dezember 2003: OpenBCM v1.04 wird freigegeben, u. a. mit Patchen von DB1RAS
- April 2004: OpenBCM v1.05, enthält u. a. FTP-Server von DH6BB

Die komplette Versionshistorie ist den gesondert erhältlichen Dokumenten *history.txt* (hier sind alle Änderungen von BCM v1.00 bis v1.46 zusammengefasst) bzw. *changes.txt* (alle Änderungen ab OpenBCM v1.00) zu entnehmen.

1.3. Von BCM zu OpenBCM

Das Projekt "Baycom-Mailbox" ist bereits schon von vielen Leuten abwechselnd mal mehr und mal weniger intensiv betreut worden. Dies ist auch nicht verwunderlich,

soll das Funkhobby doch auch ein Hobby bleiben und nicht in einen Fulltime-Job ausarten.

So ist das Grundkonzept dieser Mailboxsoftware durch Flori DL8MBT entstanden. Ab Version 1.36 hat vor allem Dietmar OE3DZW, hart am Quellcode der Mailbox gearbeitet. Immer wieder ist aber Code von Jonny DH3MB, Deti DG9MHZ und zahlreichen anderen eingeflossen.

Im Frühjahr 2000 wurde das Baybox-Projekt von DH8YMB übernommen. Es kam dann eine Phase der Neuorientierung, die vor allem DK2UI aus terminlichen Gründen übernahm, der dann auch gute Ideen realisierte.

Im September 2002 wurde mit BCM v1.46 ein Schlussstrich unter diese Phase gezogen. Das Projekt wurde in OpenBCM umgetauft und wird seitdem ausschließlich von DH8YMB betreut. Der neue Name soll nicht nur einen weiteren Neubeginn symbolisieren, vielmehr soll durch den Zusatz "Open" die Offenheit des Projektes nachhaltig unterstrichen werden.

Jeder, der Sinnvolles zu dem Projekt beitragen möchte, ist jederzeit herzlich willkommen!

1.4. Hinweise zu dieser Dokumentation

In dieser Dokumentation wird des öfteren auf Dateien verwiesen. Die Dateinamen sind dabei alle kursiv geschrieben, damit sie sich vom übrigen Text abheben.

Wichtig: Unter Linux wird zwischen den Dateien *rhosts.bcm*, *RHOSTS.BCM* bzw. *Rhosts.bcm* unterschieden; unter DOS ist dies egal. Daher ist bei Linux generell darauf zu achten, alle Dateinamen immer in Kleinbuchstaben anzugeben. Aus diesem Grund sind in dieser Dokumentation auch alle Dateinamen immer in Kleinschrift angegeben.

1.5. Versionsnummern bei OpenBCM

Bei OpenBCM wird es in Zukunft nicht mehr diese große Flut an Releaseversionen geben, wie dies bei BCM gehandhabt wurde. Neue Funktionen werden zunächst in so genannten Betaversionen implementiert und ausprobiert, die zwar ebenso von jedem getestet werden können und dürfen, jedoch immer mit dem Hinweis, dass es hier zu Problemen und Fehlern kommen kann. Diese Betaversionen werden immer mit der Versionsnummer "x.xxbx" (B=Beta) verteilt, wobei die x für einen Zahlenwert stehen. Releaseversionen werden die Versionskennung "x.xx" haben.

2. Funktionen und Grenzdaten

2.1. Grundfunktionen

- Forwarding nach W0RLI inkl. Erweiterungen von DF3AV (Lifetime, System-Mails in den Boards E (Erase) und M (MyBBS), achtstellige Bulletin-Board-Namen)
- Forwarding nach F6FBB inkl. CRC-gesichterter Komprimierung und Resume plus Erweiterungen von DL8HBS
- Unterstützung und Auswertung von hierarchischen Forwardadressen
- Unterstützung der Datei-Transfer-Protokolle AutoBIN, YAPP und Didadit

- Benutzer-Grundbefehlsumfang: DIR, LIST, READ, SEND, ERASE, REPLY, ALTER, QUIT, HELP, CHECK, MYBBS
- Üblicher Befehlssyntax, z.B. "read aktuell 1-3" oder "send netze @ dl Neuer Digi"
- Hierarchischer Aufbau der Board-Struktur
- WPROT-Unterstützung

2.2. Unterschiede zu anderen Mailboxsystemen anno 1992

Das äußere Erscheinungsbild erinnert stark an die OE5DXL-Mailbox. Dabei wird auf Kompatibilität zu anderen Mailboxen geachtet, in erster Linie zur DieBox von DF3AV. Dennoch gibt es ein paar Besonderheiten:

- Der Befehl DIR zeigt nicht wie bei DieBox und DPBox eine Liste der verfügbaren Boards an, sondern wirkt grundsätzlich wie der Befehl LIST, verfügt aber über einige Unterbefehle wie z.B. DIR BOARDS oder DIR USERS.

- Beim Befehl DIR (oder LIST) werden die Mails in der Reihenfolge aufgelistet, in der sie in der Box eingetroffen sind (nicht wie bei DXL). Um dennoch die neuesten Mails aufgelistet zu bekommen, ist die Bereichsangabe etwas anders. Beispiele:
  d 1-5 listet die ersten (also ältesten) 5 Nachrichten
  d -5 listet die letzten (also neuesten) 5 Nachrichten

2.3. Grenzdaten

Beim Design der OpenBCM-Mailbox wurde darauf geachtet, dass möglichst wenige Betriebsparameter durch interne Anschläge begrenzt werden. Überall, wo jedoch Speicherplatz nicht sinnvoll dynamisch belegt werden konnte, sondern statisch reserviert werden muss, oder wo Wertebereiche von Variablen an ihren Grenzen sind, treten Maximalwerte auf. Diese können mit dem Befehl "status limits" in der Mailbox abgefragt werden. Wegen Speichermangels sind die Limits unter DOS knapp gesetzt, unter Linux können diese Limits i. A. leicht erhöht werden.

3. Wahl der PC-Hardware und des Betriebssystems

3.1. Welcher PC?

Die Geschwindigkeit der Mailbox wird hauptsächlich durch die Festplatte, deren Controller und das verwendete Dateisystem bestimmt. Bei einem 486 DX mit 50 MHz und einer IDE-Festplatte werden unter DOS etwa 100 kB/s erreicht. Bei einem modernen, schnellen Rechner liegen unter Linux mit Extended2-Dateisystem die Werte entsprechend höher. Ein wenig spielt nur auch die Leistung der CPU eine Rolle.

Generell gilt: je schneller der PC, umso besser. Es muss aber für einen Mailbox-PC, der an einem automatischen Digipeater-Standort rumsteht, nicht der schnellste, neuste und teuerste PC einkalkuliert werden, ein guter Gebrauchter (z.B. Pentium3-Klasse) reicht heutzutage auch unter Linux oder Windows völlig aus.

Sparen kann man bei einer Mailbox außerdem bei der Grafikkarte und dem Monitor. Wer dafür eine 3D-Karte und einen 19-Zoll-Monitor verwendet, ist selbst schuld.

3.2. Festplatten-Platz

Neben der Rechner-Performance ist auch die Kapazität des Speichermediums für eine Mailbox wichtig.

Der erforderliche Plattenplatz für OpenBCM beträgt mindestens 80 MB, wenn nur User-Mails gespeichert werden sollen. Die System-Dateien (User-Daten, BID-Verwaltung, etc.) sind nicht auf Plattenersparnis ausgelegt, es sollte unbedingt ein Platz von etwa 50 MB für diese Dinge reserviert werden. Bei umfangreichen Archivboxen werden, je nach Menge an Mails und Zusatzdateien (z.B. für den Filesurf), sicherlich auch höhere Werte von bis zu mehreren Gigabyte verlangt.

All diese Werte sollten heutzutage jedoch bei den Lesern ein Grinsen auf die Lippen werfen; werden doch momentan Festplatten in den Größen von mehreren Dutzend Gigabyte verkauft. Eine solche moderne Festplatte ist also mit Sicherheit ausreichend!

Gedanken kann man sich eher machen, wie man die Datenmengen vernünftig sichert. So ist es sinnvoll, z.B. ein RAID 1 Array (Spiegelung) aus mehreren Festplatten in den Mailboxrechner einzubauen. Fällt hier eine Festplatte aufgrund eines Hardwaredefektes aus, kommt es zu keinem Datenverlust!

Als grobe Faustregel zur Festplattengröße kann folgende Tabelle betrachtet werden:

Boxgröße	Usermails	Bulletins	Lifetime User (Tage)	Lifetime Bulletins (Tage)	HDD System (MB)	HDD User (MB)	HDD Bulletins (MB)	HDD Gesamt (MB)
sehr klein	ja	nein	120	--	60	20	--	80
klein	ja	ja	360	60	70	50	400	540
mittel	ja	ja	360	180	70	100	500	700
groß	ja	ja	∞	∞	80	>500	>8000	>10000

Soll ein Digipeater auf dem gleichen Rechner mitlaufen, wird auch hierfür Speicherplatz und CPU-Leistung vom PC benötigt. Außerdem wird empfohlen, immer mindestens 500 MB auf der Platte frei zu lassen, um Backups und eine Vergrößerung der Verwaltungsinformationen jederzeit zuzulassen.

3.3. Welches Betriebssystem ist am besten geeignet?

1991 begann die Entwicklung der BayCom-Mailbox Software. Die damalige Wahl des Betriebssystems DOS geschah nach folgenden Gesichtspunkten:

- Vorhandensein einer passenden Entwicklungsumgebung (vom BayCom-Terminal und -Node)
- große Verbreitung und günstiger Preis, daher Akzeptanz zum Einsatz bei bereits laufenden bzw. neu zu installierenden Systemen
- direkter Hardwarezugriff, dadurch einfache Möglichkeiten für PR-Anbindung

Bereits zum Zeitpunkt der Entwicklung waren natürlich auch die Schwächen von DOS schon bekannt. Diese sind in erster Linie:

- Kleiner direkt adressierbarer Speicher unter 1 MB, dadurch permanente Speicherknappheit
- Segmentierung mit max. 64 kB langen linearen Speicherblöcken
- theoretisch maximal ca. 63000 Dateien pro Partition auf der Platte, praktisch noch weniger
- große Verschwendung durch das FAT-Dateisystem bei großen Platten
- schlechte Wartbarkeit, weil nur jeweils ein Programm laufen kann (kein Multitasking)

- schlechte Systemstabilität, bei Programmabsturz hängt i.A. der gesamte Rechner

Es war also im Prinzip bereits am Anfang bekannt, dass DOS nicht die ideale Wahl für den Betrieb einer Mailbox ist. Trotzdem wird es sich noch lange halten, weil das Gesetz der Trägheit und die Tatsache, dass jede Änderung mit Kosten verbunden ist, dies erzwingen.

Interessant ist die Tatsache, dass es kein anderes Betriebssystem mit ähnlich schlechten Eigenschaften wie DOS gibt. Das mag wohl daran liegen, dass DOS das erste und älteste PC-Betriebssystem ist.

Folgende Alternativen stehen potentiell zur Auswahl:

Windows 95/98/ME:
(+) Installation/Preis: einfach + bezahlbar
(-) Dateisystem: wie DOS, folglich kein Vorteil, evtl. FAT32
(+) großer Verbreitungsgrad, somit viele verfügbare Entwickler
(-) mittlerweile wird nur noch Windows ME von Microsoft supported

OS/2:
(-) Installation/Preis: einfach + bezahlbar
(o) Dateisystem: HPFS, besser als DOS-FAT
(-) hohe Hardwareanforderungen im Vergleich zu DOS durch graphische Oberfläche, besser Systemstabilität als DOS
(-) wird von IBM nicht mehr supported

Windows NT/2000/XP:
(+) Installation/Preis: einfach, aber recht teuer
(o) Dateisystem: NTFS, weit besser als DOS-FAT
(-) hohe Hardwareanforderungen im Vergleich zu DOS durch graphische Oberfläche, aber hohe Systemstabilität

Linux:
(+) Installation: je nach Distribution von einfach bis umständlich und fehlerträchtig, dafür aber kostenlos
(+) Dateisystem: ext2-Dateisystem sehr schnell und effizient
(+) graphische Bedienoberfläche nicht erforderlich, daher vergleichbar mit DOS, System offen

Andere UNIX-Derivate für PC wurden nicht betrachtet, da der Preis meist zu hoch oder die Beschaffung schwierig ist.

Unter dem Strich fallen die Betriebsysteme OS/2 und Windows95/98/ME schon auf Grund des nicht mehr gewährleistetem Support aus.

Als DOS-Alternativen bleiben somit Windows NT/2000/XP und Linux über. Es soll hier nicht zur Diskussion angeregt werden, welches der genannten Systeme nun das Beste ist und welches nicht, trotzdem zeigt sich auf Grund der Vor-/Nachteile, das das Linux-System wohl als am ehesten geeignet ist, den Ansprüchen eines Mailbox-Systems gerecht zu werden.

3.3.1. Linux

3.3.1.1. Warum Linux?

Wie auch immer die Vor- und Nachteile der einzelnen Systeme ausfallen mögen, die erste Portierung der OpenBCM-Mailbox wurde für Linux durchgeführt.

Die ersten Schritte dieser Portierung haben schon Mitte 1994 stattgefunden. Es war weniger die unbedingte Notwendigkeit, als vielmehr der technische Reiz, der zu diesem Schritt geführt hat. So ist es auch nicht verwunderlich, dass nach den ersten Erfolgserlebnissen dieser Zweck erfüllt war, und das Projekt wieder in die Ecke geworfen wurde.

Einige Leute haben sich mit der damaligen BCM Version 1.35a "vergnügt". Diese war allerdings für den praktischen Einsatz vollkommen unbrauchbar und vermochte lediglich das Erlebnis "es geht prinzipiell" zu demonstrieren.

Im Dezember 1995 brach dann erneutes Interesse aus und das Projekt wurde aus der tiefen Versenkung ausgegraben. Da der Code größtenteils portabel ist, konnte auf die zwischenzeitliche Weiterentwicklung der DOS-Version problemlos aufgesetzt werden.

Komplexere Betriebssysteme haben zumeist die Eigenschaft, auf einer großzügig bemessenen Hardware sehr viel bessere Leistungseigenschaften zu besitzen als z.B. DOS. Das betrifft sowohl die Dateizugriffe als auch den Umgang mit Arbeitsspeicher. Ist der Rechner jedoch unangemessen klein, so brauchen die Mechanismen innerhalb des Betriebssystems einen großen Anteil der zur Verfügung stehenden Resourcen und bewirken daher keine Leistungssteigerung, sondern eher ein Ausbremsen der Anwendung. Dies gilt natürlich sowohl für Linux als auch für Windows.

Die ersten Linux-Gehversuche Anfang 1996 bei DB0AAB-8 waren erfreulich. Den Anforderungen bezüglich Rechnerleistung wurde dort Rechnung getragen. Dort wurde seinerzeit ein 486DX2/66 mit 16 MB RAM und insgesamt 4,5 GB Plattenspeicher eingesetzt. Heutzutage ist gilt dies natürlich schon wieder als völlig veraltert. Es sollte sich hier auch zeigen, wie sich die Box im Zusammenhang mit einer sehr hohen Datenmenge verhält. Aus diesem Grund ist auch derzeit und bis auf weiteres jegliches Löschen von Nachrichten bei DB0AAB abgestellt. Falls die vorhandenen Platten voll sind, lässt sich die Kapazität problemlos auf höhere Kapazitäten erweitern.

Die Linux-Version der Mailbox hat vor dem ersten Einsatz im "rauhen" Betrieb bei DB0AAB auf der Fachhochschule München viele Versuche im stillen Kämmerlein über sich ergehen lassen. Trotzdem war es natürlich nicht möglich, alle Eventualitäten des praktischen Betriebs auszutesten, jedoch lief die Mailbox dort schon recht zügig auch unter höheren Belastungen sehr stabil. Inzwischen hat sich der Betrieb normalisiert und es gibt auch kein auffallendes Feedback von den Benutzern. Dies ist ein typisches Merkmal bei jeder technischen Entwicklung, dass Rückmeldungen stets nur im Fehlerfall kommen. Das ist zwar für den Entwickler nicht besonders erfreulich, aber im Sinne der Fortentwicklung durchaus wünschenswert und notwendig.

Generell gilt: Linux ist ideal über Funk zu bedienen und hat insgesamt Eigenschaften, die geradezu ideal für einen Einsatz der OpenBCM-Mailbox sind. Linux hat eigentlich nur einen einzigen Nachteil: Wer sich nicht mit UNIX auskennt, braucht ein relativ großes Grundwissen, um alle nötigen Kenntnisse zum Betrieb und zur Wartung eines Rechners zu haben.

3.3.1.2. Hardware-Anforderung unter Linux

Hier wird mindestens ein Rechner mit 80486-Prozessor bei 33 MHz, 8 MB RAM (besser 16 MB) und 1 GB Festplatte empfohlen. Mit schlechterer Hardwareausstattung ist die DOS-Version zu bevorzugen. Die Linux-Version sollte möglichst auf dem Mailbox-Rechner compiliert werden, um Schwierigkeiten mit inkompatiblen C-Bibliotheken zu vermeiden. Entsprechende Programme und Tools (C++-Compiler, Bibliotheken, etc.) sind bei jeder Linux-Distribution enthalten. Ggf. kann aber auch ein statisch gelinktes Binary verwendet werden.

3.3.2. Windows

3.3.2.1. Warum Windows?

Für eine Anwendung wie eine PR-Mailbox ist Windows eigentlich kein allzu geeignetes Betriebssystem. Was die Vor- und Nachteile der einzelnen Systeme sind, sollte hier nicht diskutiert werden.

Tatsache ist, dass unter DOS nach wie vor nur die berühmten 640 kB Speicher vernünftig ansprechbar sind, und dass unter DOS kein besonders herausragendes Dateisystem zur Verfügung steht.

Windows NT/2000/XP hilft beiden Problemen ab. Es steht eine virtuelle Speicherverwaltung und das Dateisystem NTFS zur Verfügung.

Es ist nicht zu verschweigen, dass technisch die Lösung für Linux die weitaus bessere ist. Nur: Wer sich nicht mit UNIX auskennt, braucht ein relativ großes Grundwissen, um alle nötigen Kenntnisse zum Betrieb und zur Wartung eines Linux-Rechners zu haben. Windows ist hier ein wenig besser, vor allem ist das erforderliche Wissen etwas weiter gestreut. Dies ist die einzige Motivation, warum es die OpenBCM-Mailbox für Windows gibt. Es ist auch bei langem Nachdenken keine technische Begründung zu finden, warum Windows besser sein sollte als Linux, es ist eben nicht so. Da die Windows-Version der Mailbox nur wenig verbreitet ist, muss auch mit mehr bisher unentdeckten Fehlern als bei Linux gerechnet werden.

Die Windows-Version verhält sich sehr viel ähnlicher zur Linux- als zur DOS-Version. Das ist auch kein Wunder, da im 32 Bit-Mode vieles aus der DOS-Welt (z.B. Hardware- und BIOS-Zugriffe) nicht zur Verfügung steht, und auch z.B. die Speicherverwaltung eher vergleichbar mit Linux ist.

3.3.2.2. Welches Windows: NT/2000/XP oder Win95/98/ME?

Klare Antwort: Nur Windows NT/2000/XP! Die *bcm32.exe* läuft nur bedingt unter Windows 95/98/ME! Das liegt mehr oder weniger daran, dass man ohne Aufwand eine für beide Systeme kompatible EXE-Datei erzeugen kann, zum Betrieb mit einer Mailbox scheidet Windows 95/98/ME jedoch schon deshalb aus, weil es kein dem DOS überlegenes Dateisystem gibt. Mittlerweile ist Win95/98 und auch WinME als Betriebsystem langsam überholt und selbst Microsoft forciert die Nutzung eines stabileren Betriebsystems wie Windows XP. Unter Windows NT/2000/XP ist ausschließlich ein Betrieb mit NTFS-Dateisystem ratsam. Die Windows-Version der OpenBCM-Mailbox, BCM32, wurde ursprünglich unter Windows NT 4.0 mit Service Pack 5 entwickelt, mittlerweile wird unter Windows XP daran gebastelt.

3.3.2.3. Hardware-Anforderung unter Windows (NT/2000/XP)

Hier gelten dieselben Anforderungen an die Hardware wie unter Linux: mindestens ein schneller 486 oder ein PC der Pentium-Klasse sollte es sein. Zusätzlich sollte jedoch beachtet werden, dass Windows auf Grund der grafischen Oberfläche sehr viel verschwenderischer mit RAM umgeht als Linux. Unter WinNT sollten deshalb mindestens 32 MB, unter Win2000 oder WinXP mindestens 128 MB, besser 256 MB verwendet werden.

3.3.3. DOS

Zu DOS ist an dieser Stelle eigentlich nicht mehr viel zu erwähnen. Da das Betriebssystem DOS mittlerweile hoffnungslos veraltert ist, ist keinem Neueinsteiger zu empfehlen, diese Version zu installieren. Ggf. macht es bei Sinn, bestehende Systeme unter DOS weiterlaufen zu lassen, jedoch immer mit den Einschränkungen die dieses Betriebssystem mit sich bringt (640 kB Hauptspeicherbarriere, fehlerträchtiges FAT-Dateisystem, kein TCP/IP-Support...). Aber selbst hier macht es Sinn ein klein wenig Aufwand zu investieren, um die bestehende Mailbox auf Linux oder Windows umzurüsten (siehe hierzu auch Kapitel "21. Umstellung DOS-Version auf Windows oder Linux").
DOS – Ruhe in Frieden!

3.3.3.1 Hardware-Anforderung unter DOS

Um die geforderte Leistungsfähigkeit sicherzustellen, werden unter DOS folgende Eigenschaften vom verwendeten Rechner erwartet:

- Mindestens ein Rechner mit 80386-Prozessor
- Platte mit max. 20 ms Zugriffszeit und mindestens 500 kB/s Durchsatz
- Verwendung eines Platten-Cache Programms - deshalb sollte der Rechner mit mindestens 2 MB RAM bestückt sein (das entspricht dann 1,3 MB Cache), besser sind 4 MB; der Aufruf mit Standardeinstellung ist sinnvoll, also einfach in der AUTOEXEC.BAT "SMARTDRV 4000" eintragen (bei 4 MB RAM)
- Empfohlen wird MS-DOS 5.0 oder 6.2, mindestens jedoch DOS 3.3; abgeraten wird von DR DOS 5.0 oder MSDOS 6.0; auch der Einsatz von OpenDOS ist möglich (dieses ist für den privaten Einsatz kostenlos verfügbar). OpenDOS ist im Internet unter z.B. http://www.opendos.de/ zu finden.

3.4. Fazit

Als Fazit des Betriebssystem- und OpenBCM-Varianten-Vergleich gilt:
Linux ist sicherlich die beste Wahl. Gegebenfalls kann man sich auch die Windows-Variante der OpenBCM antun. Von DOS ist heutzutage generell abzuraten!

Die Wahl der PC-Hardware spielt eigentlich kaum noch eine Rolle, hier ist quasi jeder PC größer 486 in der Lage eine Mailbox vernünftig zu hosten, egal für welches Betriebssystem man sich entscheidet.

4. Installation

4.1. Bezug der vollständigen Distribution der OpenBCM-Mailbox

Für die Erstinstallation ist es sinnvoll, zunächst eine vollständige Distribution der Box zu besitzen. In diesem Komplettarchiv sind dann alle Dateien enthalten plus diverse Beispieldateien als Konfigurationsbeispiel.

Um die Box unter DOS zum Laufen zu bringen sind minimal 2 Dateien notwendig:

- *bcm.exe* (Das Mailbox-Programm selbst)
- *msg/messages.gb* (Die Sprach-Datei)

Um die Box unter Linux oder Windows zum Laufen zu bringen, ist nur eine Datei minimal notwendig, die Sprachdatei wird hier automatisch erzeugt:

- *bcm32.exe* (für Windows)

- *bcm* (für Linux)

Ein Verzeichnis aller Dateien der Mailbox befindet sich weiter hinten in dieser Beschreibung. Der Großteil dieser Dateien, sowie alle Verzeichnisse werden von der Mailbox im Betrieb automatisch erzeugt.

Eine jeweils aktuelle Version der kompletten OpenBCM-Mailbox-Distribution (rund 8,5 MB) ist im Amateurfunk-PR-Netz im Filesurf/FTP-Server von DB0ZWI in Zwickau, oder im CB-PR-Netz im Filesurf bei DBO274 in Oelde zu finden. Manche andere Boxen haben diese Dateien ebenfalls übernommen, manchmal lohnt sich also auch eine Suche nach Mails mit "fobcm" im Titel in der eigenen Heimatmailbox. Wer einen Internet-Zugang hat, kann auf der Webseite http://dnx274.dyndns.org/baybox immer die aktuelle Version und diverse andere Dateien für die OpenBCM-Mailbox finden.

4.2. Bezug von Updates in Mailboxen

Die jeweils aktuelle Update-Version, alle verfügbaren Beta-Versionen und natürlich auch die komplette Dokumentation sind

...im Amateurfunk-PR-Netz im Filesurf/FTP-Server von DB0ZWI in Zwickau,
...im CB-PR-Netz im Filesurf von DBO274 in Oelde,
...im Internet unter http://dnx274.dyndns.org/baybox zu finden.

Sicherlich gibt es auch noch viele andere Mailboxen, die die Software anbieten. Im Packet-Radio-Netz wird die Rubrik BAYBOX@BAYCOM genutzt, um über alle Belange der Software zu diskutieren. Dort werden auch alle Release-Versionen eingespielt und veröffentlicht.

4.3. Einrichtung der Festplatte

4.3.1. Unter DOS

Die Mailbox legt für jede Mail eine eigene Datei an. Da im DOS FAT16-Dateisystem nur max. 65535 Cluster je Partition zur Verfügung stehen, ergibt sich bereits mit einer 540 MB-Platte ein Speicherverbrauch von mindestens 8 kB je Datei (sofern nur eine Partition eingerichtet wurde). Es ist deshalb sinnvoll für Usermails, Bulletins und das System jeweils eine eigene Partition einzurichten.

Seit Windows 95B besteht auch die Möglichkeit bei einer Platte mit mehr als 500 MB Kapazität eine 32 Bit-FAT einzurichten. Der Vorteil ist, dass pro Datei nur minimal 4 kB verbraucht werden und es sind auch Partitionen größer 2 GB möglich. Zu beachten sind dabei die lizenzrechtlichen Bestimmungen der Softwarefirma. Allerdings gibt es mit der dazu notwendigen DOS-Version Probleme beim DOS-Austieg.

Bei einer 540 MB-Platte empfiehlt sich folgende Konfiguration:

```
Laufwerk C: - erste Partition:  120 MB - System
Laufwerk D: - zweite Partition: 300 MB - Bulletins
Laufwerk E: - dritte Partition: 120 MB - Usermails
```

(Bei größeren Platten mit dem entsprechenden Faktor multipliziert.)

4.3.2. Unter Linux

Es sollte auf jeden Fall mindestens das Dateisystem "ext2" verwendet werden. Prinzipiell ist auch der Betrieb mit jedem anderen Dateisystem möglich. Bei der Verwendung des DOS-Dateisystems "(V)FAT" ist jedoch mit großen Geschwindigkeitseinbussen zu rechnen.

Die Blockgröße sollte bei der Bulletin- und Usermail-Partition 1024 Bytes betragen.

4.3.3. Unter Windows

Auch hier ist das Dateisystem "(V)FAT" von DOS möglichst zu meiden. Windows NT/2000/XP bietet das Dateisystem "NTFS", dies sollte dann auch verwendet werden.

4.4. Installation unter DOS

Die Mailbox kann so ziemlich unter jeder Konfiguration des Rechners laufen. Für einen sinnvollen Boxbetrieb sollten jedoch folgende Punkte erfüllt werden:

- Aktuelle DOS-Version (MSDOS 5.0 oder 6.2, IBMDOS 5.0 oder 6.3, Win95/98-DOS)
- Optimierte *config.sys*
- Optimierte *autoexec.bat*
- Sinnvolle Partitionierung der Platte

Zu einem ersten Start der Mailboxsoftware unter DOS sind mindestens zwei Dateien notwendig, und zwar *bcm.exe* (das DOS-Executable) und die Datei *msg/messages.gb* (englischsprachige Boxtexte).

Alle anderen zum Betrieb der Mailbox notwendigen Dateien und Verzeichnisse können von der Mailbox selbst erzeugt werden. Die Mailbox kann nun durch die Eingabe von "BCM <Enter>" gestartet werden. Damit sind erste Tests möglich.

An der Konsole kann jetzt ein Rufzeichen zum Login eingegeben werden, die Bedienung der Mailbox erfolgt anschließend wie über Funk gewohnt.

Durch Drücken der Tastenkombination ALT-X wird die Mailbox beendet.

Um die Mailbox wirklich "on air" betreiben zu können, müssen diverse Dateien angepasst werden (siehe dazu auch Kapitel "5. Konfiguration für den ersten Start" und folgende).

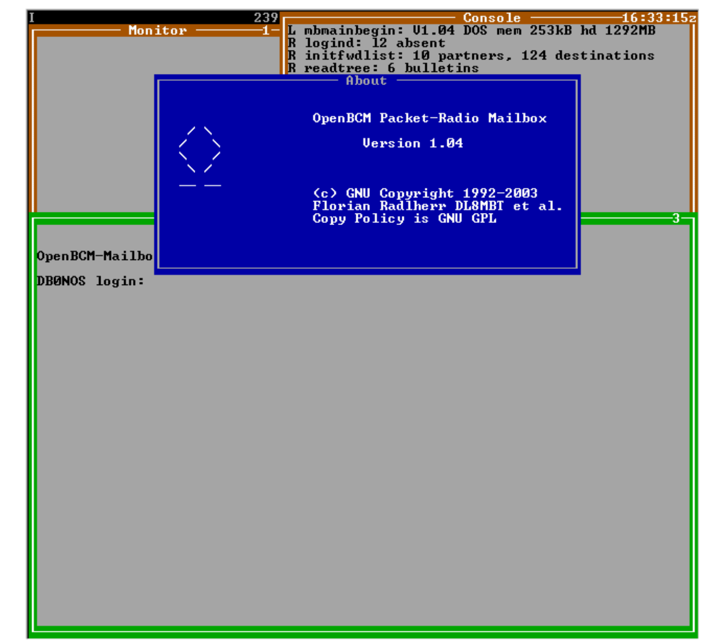


Abb. rechts: Blick auf die Bedienoberfläche unter DOS

4.4.1. Optimierte *config.sys*

In der Datei *config.sys* sollte ein Wert FILES=50 eingestellt werden. Außerdem sollte DOS auf maximalen freien konventionellen Speicher konfiguriert sein

(mittels DOS=HIGH, EMM386, etc.). Nicht wirklich für den Mailboxbetrieb notwendige Gerätetreiber sollten nicht geladen werden, notwendige mit "DEVICEHIGH=" in den hohen Speicherbereich. Sinnvoll ist der Betrieb auf einer Monochrom-Grafikkarte ("Hercules") oder auf einer CGA-Karte, möglichst nicht EGA oder VGA, da man hierdurch mehr Arbeitsspeicher gewinnt. Sofern es das BIOS zulässt, ist auch ein Betrieb ohne Grafikkarte möglich.

Beispiel aus einer *config.sys* für Monochrom-Karten:
DEVICE=C:\DOS\EMM386.EXE NOEMS I=A000-AFFF I=B800-EFFF
Beispiel aus einer *config.sys* für EGA/VGA-Karten:
DEVICE=c:\DOS\EMM386.EXE NOEMS I=C800-EFFF

Werden SCSI-, RLL- oder ESDI-Plattencontroller verwendet, so ist die Option "X=C800-CFFF" und ein Aussparen dieses Bereichs in den "I="-Anweisungen erforderlich, da diese Contoller in diesem Bereich ein BIOS haben.

Beispiel einer vollständigen *config.sys*:
```
LASTDRIVE=Z
DEVICE=C:\DOS\HIMEM.SYS
DOS=HIGH,UMB
BUFFERS=50,3
FILES=60
DEVICEHIGH=C:\DOS\EMM386.EXE NOEMS I=A000-AFFF I=B800-EFFF
REM For monochrom video only I=A000-AFFF
SHELL=C:\DOS\COMMAND.COM /P
```

4.4.2. Optimierte *autoexec.bat*

Wenn die Mailbox fix auf einem Rechner installiert wird, sollte sie in der Datei *autoexec.bat* in einer Endlosschleife aufgerufen werden. Als Tastaturtreiber sollte nur der Original-DOS-Treiber *keyb.com* verwendet werden. Andere Tastaturtreiber führen oft zu sehr merkwürdigen Effekten, auch wenn sie Arbeitsspeicher einsparen.

Beispiel einer *autoexec.bat*:
```
@ECHO OFF
SET TEMP=C:\TEMP               *** Verzeichnis muss existieren!
SET COPYCMD=/Y                 *** für MSDOS 6.x: Keine Rückfragen beim copy
SET FLEXNET=C:\FLEXNET         *** Pfad für PC/FlexNet einstellen
PATH C:\DOS;C:\BCM;C:\BCM\RUN  *** Im Suchpfad sollten sich nur
                                   wirklich notwendige Pfade befinden
PROMPT $P$G                    *** Übliches Befehlsprompt
LOADHIGH SHARE.EXE             *** Regelt Zugriffsrechte auf der Platte
LOADHIGH SMARTDRV 4000         *** Plattencache in den hohen Speicherbereich
LOADHIGH KEYB GR               *** Deutscher Tastaturtreiber
:LOOP
CHOICE /T:j,5 Knoten starten?  *** MSDOS 6.x: Abfrage ob Knoten starten
IF ERRORLEVEL=2 GOTO ENDE
ECHO Knoten wird gestartet...
CALL SFLEX.BAT                 *** Vor der Mailbox die FlexNet-Treiber starten
ECHO Knoten aktiv
SCANDISK /ALL /AUTOFIX /NOSAVE /NOSUMMARY /MONO C:
                               *** Platte ausmisten (MSDOS 6.x),
CD \BCM
ECHO Box wird gestartet...
BCM                            *** Startet die Mailbox
CALL STOPFLEX                  *** Beendet FlexNet
GOTO LOOP                      *** Wenn BCM beendet, dann wieder zurück
:ENDE
```

Anmerkung: Beim Aufruf von *smartdrv* sollte in der Kommandozeile sämtlicher vorhandener XMS-Speicher angegeben werden. Die OpenBCM-Mailbox nutzt XMS-Speicher nur als Plattenpuffer, eine anderweitige Nutzung ist nicht möglich. Auch dies bringt allerdings einen sehr großen Geschwindigkeitsvorteil.

Es sollte darauf geachtet werden, dass nach dem Start der OpenBCM-Mailbox noch mindestens 100 kB, besser aber mehr als 250 kB konventioneller Speicher frei sind. Dies ist normalerweise zu erreichen.

4.5. Installation unter Linux

4.5.1. Systemvoraussetzungen unter Linux

Voraussetzung ist Linux 2.0.0 oder höher. Es sollte ein möglichst stabiler Kernel verwendet werden, von Hackerversionen wird abgeraten. Gute Erfahrungen bestehen mit den derzeit üblichen 2.2.x- und 2.4.x-Kerneln. ELF-Unterstützung und die dazugehörigen Bibliotheken sind erforderlich.

4.5.2. Grundinstallation unter Linux

- Im Regelfall wird man unter Linux zunächst die Mailboxsoftware selber kompilieren wollen/müssen. Dies ist ausführlich im Kapitel "26.2. Unter Linux" beschrieben.
- Die Installation des Box-Programms erfordert root-Rechte. Es muss sich jeder im Klaren sein, dass sich durch unüberlegte Handlungen im System viel kaputt machen lässt. Jegliche Haftung ist natürlich ausgeschlossen. Folgende Vorgehensweise ist ein Vorschlag. Wer Erfahrung hat kann natürlich nach Gutdünken davon abweichen. Zur allgemeinen Einheitlichkeit macht es jedoch durchaus Sinn, dass es jeder in etwa gleich macht.
- Es sollte ein User und eine Gruppe "bcm" eingerichtet werden. UID und GID sind im Prinzip egal. Bei vernetzten Systemen sollten sie (müssen aber nicht) einheitlich sein. Wenn sonst keine Präferenzen bestehen, wird UID 666 und GID 66 vorgeschlagen. Bei manchen Distributionen (etwa Redhat) wird die GID 66 bereits anderweitig verwendet. Es führt zu keinen weiteren Problemen, wenn eine andere GID verwendet wird. Das Einrichten eines speziellen Users macht die Regelung der Zugriffsrechte einfacher. Als Loginshell kann "bct" angegeben werden, dann wird bei einem Login des Users "bcm" direkt das externe Terminalprogramm *bct*, was ebenfalls erhältlich ist, aufgerufen.
- Die Installation kann z.B. im Verzeichnis */bcm* erfolgen. Ein anderes Verzeichnis ist über die Enviroment-Variable $BCMHOME einstellbar. Sie muss nicht existieren, wenn obiges Verzeichnis verwendet wird. Empfohlen wird die Verwendung des Skripts *startbcm*. *startbcm* ist ein Shell-Script, das BCMHOME setzt, ins richtige Verzeichnis verzweigt und *bcm* startet.
- Wenn eine Box unbemannt betrieben werden soll, ist ein Eintrag in */etc/inittab* empfehlenswert. Der Eintrag lautet sinngemäß:
  ```
  bc:23:respawn:/bcm/startbcm
  ```
  Durch diesen Eintrag wird die Box nach dem Booten automatisch gestartet und wird auch neu gestartet, falls sie abstirbt oder mit dem Remote-Befehl SHUTDOWN vom Sysop beendet wird. Nach Ausführung des Skriptes *startbcm* wird eine Weile gewartet, damit im Fehlerfall der respawn-Vorgang nicht zu schnell abläuft. Der Runlevel (die Zahlen nach bc:) muss zur System-Installation passen. Es gibt verschiedene Versionen, es muss jedenfalls der Zustand "Multiuser mit Netzwerk" erreicht sein.
- Wenn alle Dateien installiert sind (und ggf. von der DOS-Version kopiert und konvertiert sind), sollten diese dem User "bcm" und der Gruppe "bcm" gehören. Sicherheitshalber sollte also im *bcm*-Verzeichnis und ggf. im *user*- und *info*-Verzeichnis folgendes gemacht werden:
  ```
  chown -R bcm:bcm *
  chmod -R u+rw *
  ```
- Da die Box sinnvollerweise nicht als root läuft, ist für den Sysop ein root-Login erforderlich. Dies wird über das Programm *pw* gemacht (siehe weiter unten). Das Programm muss allerdings bei der Installation root-

Rechte und setuid root erhalten. Also muss folgendes gemacht werden:

```
chown root.root pw
chmod 6777 pw
```

Erst dann ist *pw* geeignet, einen Sysop zum root zu machen. Alternativ dazu ist es auch möglich, eine zweite Instanz der OpenBCM-Mailbox mit root-Rechten laufen zu lassen, diese mit einem anderen Passwort zu versehen und über diese dann die Systemwartung durchzuführen. Beim Start einer zweiten Instanz muss mit "TELNET_PORT <port>" in *init.bcm* ein anderer TCP-Port angegeben werden und auch in *init.l2* ein anderer UDP-Port bzw. eine anderer Schnittstelle zur Kommunikation verwendet werden, um Kollisionen zu verhindern.

```
Telnet server
31.03.04 15:09:36z SYSTEM: #R working dir: '/bcm'
31.03.04 15:09:36z SYSTEM: #R l1axip: 192.168.0.5 rxport=97 txport=97
31.03.04 15:09:36z SYSTEM: #R l1axip: initialised to channel 0
31.03.04 15:09:36z SYSTEM: #R l1axip: 192.168.0.2 rxport=96 txport=96
31.03.04 15:09:36z SYSTEM: #R l1axip: initialised to channel 1
31.03.04 15:09:36z SYSTEM: #L Start: V1.05b21 Linux mem 289976kB hd 37008MB
31.03.04 15:09:36z SYSTEM: #R init_tnc: 200 ports
31.03.04 15:09:36z SYSTEM: #R ax25k_init: Kernel AX25 disabled
31.03.04 15:09:36z SYSTEM: #R initfwdlist: 21 partners, 181 destinations
31.03.04 15:09:36z SYSTEM: #R initafwdlist: 18 partners
31.03.04 15:09:36z SYSTEM: #R readtree: 283 bulletins
31.03.04 15:09:36z SYSTEM: #R cvtload: 1080 convert
31.03.04 15:09:36z SYSTEM: #R runutils: 27 runutils activated
31.03.04 15:09:36z SYSTEM: #R telnetd: initialised to port 4719
31.03.04 15:09:36z SYSTEM: #R httpd: initialised to port 8080
31.03.04 15:09:36z SYSTEM: #R ftpd: initialised to port 8021
31.03.04 15:09:36z SYSTEM: #R sendmail: initialised to port 8025
31.03.04 15:09:36z SYSTEM: #R popper: initialised to port 8110
31.03.04 15:09:36z SYSTEM: #R nntpd: initialised to port 8119
31.03.04 15:09:36z SYSTEM: #R radiod: radio connectivity disabled
31.03.04 15:09:36z SYSTEM: #R serv: initialised to port 8123
31.03.04 15:09:37z SYSTEM: #R mbcron: started
```

Abb. oben: Screenshot der OpenBCM unter Linux

Um die Mailbox wirklich "on air" betreiben zu können, müssen diverse Dateien angepasst werden (siehe dazu Kapitel "5. Konfiguration für den ersten Start" und folgende).

4.5.3. Ansteuerung über TCP/IP unter Linux

Die OpenBCM-Mailbox lauscht defaultmäßig am Telnet-Port 4719. Das Terminal *bct* greift nur dann auf diesen Port zu, wenn in */etc/services* ein Service mit Namen "bcm" eingerichtet wird:

```
bcm 4719/tcp # OpenBCM-Mailbox
```

Für UDP muss der Port nicht unbedingt eingerichtet werden, es schadet jedoch nicht. Die Portnummer 4719 ist frei erfunden und sollte im Sinne der Einheitlichkeit so verwendet werden, wenn keine Kollisionen mit anderen Applikationen auftreten. Die Portnummer kann mit dem Parameter "TELNET_PORT <port>" in der Datei *init.bcm* auch anders eingestellt werden. Wird dort 0 eingestellt, so ist der Telnet-Port inaktiv.

4.5.4. Einloggen in die Box unter Linux

Da die OpenBCM-Mailbox unter Linux selbst keine Benutzeroberfläche hat, wird über ein externes Programm auf die Box zugegriffen. Das mitgelieferte Programm heißt *bct* und ist ein primitives Terminalprogramm mit dem man derzeit ausschließlich eine Verbindung zur Box selbst herstellen kann. Während das Programm läuft, darf ggf. die Fenstergröße nicht verändert werden weil dies nicht abgefangen wird. Beendet wird durch einen Logout mit "q" bzw. durch

CTRL-C. Der Aufruf zum Start des Terminalprogramms lautet:

```
bct [<hostname>]
```

Die Angabe des Hosts kann entfallen, wenn der Login auf demselben Rechner erfolgt. Der Login kann jedoch von einem beliebigen Rechner aus erfolgen. Nach Start des Terminals wird ein Loginprompt ausgegeben. Auf dem Rechner, auf dem *bct* laufen soll, muss der oben beschriebene Eintrag in */etc/services* erfolgt sein!

Erfolgt der Login von einem Rechner aus, der in der Datei *rhosts.bcm* angegeben ist, so wird beim Login nur das Rufzeichen verlangt und der eingeloggte Benutzer ist danach automatisch Sysop in der Box. Siehe hierzu auch das Kapitel "5.7. Remotehost (*rhosts.bcm*)".

Zum Login in die Box kann auch Telnet verwendet werden. Allerdings ist hierbei mit Gebrechen bei der Korrektur einer Kommandozeile zu rechnen, da diese andernfalls im Terminalprogramm und nicht in der Box erfolgt. Der Aufruf lautet hier:

```
telnet <hostname> bcm
```

oder, wenn kein Eintrag in */etc/services* erfolgt ist:

```
telnet <hostname> 4719
```

4.6. Installation unter Windows

4.6.1. Systemvoraussetzungen unter Windows

Als Windows-Betriebssystem kommt nur Windows NT, 2000 oder XP in Frage, nicht jedoch Windows 95/98/ME! Zusätzlich muss für Windows das TCP/IP-Protokoll im Windows-Netzwerk installiert sein. Ohne diese Installation funktioniert die Windows-Version der OpenBCM-Mailbox nicht.

4.6.2. Grundinstallation unter Windows

Zu einem ersten Start der Mailboxsoftware ist mindestens die Datei *bcm.exe* (das Windows-Executable) nötig. Diese sollte man in einen leeren Ordner kopieren, vorzugsweise *c:\bcm*.

Beim ersten Aufruf werden diverse Unterverzeichnisse und Konfigurationsdateien von der Mailbox selber erzeugt, die im weiteren Betrieb benötigt werden. Damit sind erste Tests möglich.

Per Telnet oder Terminalprogramm via Flex32/XNET-Anbindung kann die Mailbox anschließend, wie über Funk gewohnt, bedient werden.

Durch Drücken der Tastenkombination ALT-X wird die Mailbox beendet.

Um die Mailbox wirklich "on air" betreiben zu können, müssen diverse Dateien angepasst werden (siehe dazu Kapitel "5. Konfiguration für den ersten Start" und folgende).

Damit die Mailbox-Software unter Windows nach einem (normalerweise nicht vorkommenden Absturz) automatisch neu startet, gibt es ein kleines Software-Tool mit dem Namen "OpenBCM Watchdog". Dies sollte man vor allem dann installieren, wenn man die Mailbox an exponierter Stelle im unbeaufsichtigten Betrieb nutzen will.

```
C:\bcmnt\bcm32.exe
OpenBCM V1.05 (WIN32)(c) GNU GPL 1992-2004 F. Radlherr, DL8MBT et al.
OpenBCM is maintained by Markus Baumann, DH8YMB
http://www.baycom.org/baybox/
31.03.04 17:24:05l SYSTEM: #R working dir: 'C:/bcmnt'
31.03.04 17:24:05l SYSTEM: #R ws_init: Sockets 32767, UDP size 65467
31.03.04 17:24:05l SYSTEM: #R ws_init: Sockets Version: 1.1 - High Version: 2.2
31.03.04 17:24:05l SYSTEM: #R ws_init: WinSock 2.0 Running
31.03.04 17:24:06l SYSTEM: #R l1axip: 192.168.0.5 rxport=129 txport=129
31.03.04 17:24:06l SYSTEM: #R l1axip: initialised to channel 0
31.03.04 17:24:06l SYSTEM: #R l1axip: 127.0.0.1 rxport=4866 txport=4867
31.03.04 17:24:06l SYSTEM: #R l1axip: initialised to channel 1
31.03.04 17:24:07l SYSTEM: #L Start: V1.05 WIN32 mem 88904kB hd 24610MB
31.03.04 17:24:07l SYSTEM: #R init_tnc: 200 ports
31.03.04 17:24:07l SYSTEM: #R initfwdlist: 1 partners, 22 destinations
31.03.04 17:24:07l SYSTEM: #R readtree: 116 bulletins
31.03.04 17:24:07l SYSTEM: #R cvtload: 1071 convert
31.03.04 17:24:07l SYSTEM: #R telnetd: initialised to port 7719
31.03.04 17:24:07l SYSTEM: #R httpd: initialised to port 7080
31.03.04 17:24:07l SYSTEM: #R ftpd: initialised to port 8021
31.03.04 17:24:07l SYSTEM: #R sendmail: initialised to port 7025
31.03.04 17:24:07l SYSTEM: #R popper: initialised to port 7110
31.03.04 17:24:07l SYSTEM: #R nntpd: initialised to port 7119
31.03.04 17:24:07l SYSTEM: #R serv: initialised to port 7123
31.03.04 17:24:08l SYSTEM: #R mbcron: started
```

Abb. oben: Blick auf die Oberfläche von OpenBCM unter Windows NT/2000/XP

4.6.3. Ansteuerung über TCP/IP unter Windows

Die OpenBCM-Mailbox lauscht defaultmäßig am Telnet-Port 4719. Die Portnummer 4719 ist frei erfunden und sollte im Sinne der Einheitlichkeit so verwendet werden, wenn keine Kollisionen mit anderen Applikationen auftreten. Die Portnummer kann mit dem Parameter "TELNET_PORT <port>" in der Datei *init.bcm* auch anders eingestellt werden.

Wird dort als Wert 0 eingestellt, so ist der Telnet-Port inaktiv.

4.6.4. Einloggen in die Box unter Windows

Da die OpenBCM-Mailbox unter Windows selbst keine Benutzeroberfläche hat, kann nur über ein externes Programm auf die Box zugegriffen werden.

Zum Login in die Box kann das Windows-Programm Telnet verwendet werden. Allerdings ist hierbei mit Gebrechen bei der Korrektur einer Kommandozeile zu rechnen, da diese andernfalls im Terminalprogramm und nicht in der Box erfolgt.

Der Aufruf lautet hier (z.B. unter START/AUSFÜHREN eingetippt):

```
telnet localhost 4719
```

Ferner kann man natürlich auch externe Terminalprogramme wie z.B. Paxon, Winstop oder WPP, die alle per Flex32 oder XNET über einen internen AXUDP Link an die Mailbox angebunden sind, verwenden, um mit der Mailbox kommunizieren können.

4.7. Update auf eine neue Software-Version

Unter DOS wird ein Update der Mailboxsoftware am besten über eine Batch-Datei durchgeführt. Es lohnt sich, folgende Batch-Datei *newbcm.bat* (EXTRACT in *config.h* muss in die Mailbox einkompiliert sein, oder das Sysop-Tool *extract.exe* von DF3VI muss vorliegen) anzulegen:

```
extract baybox %1.p0
7plus %1.p01
copy bcm.exe bcm.alt
if exist %1.exe copy %1.exe bcm.exe
del %1.*
```

Falls 7plus Fehler meldet, muss man ggf. eine *.cor*-Datei anfordern.

Das eigentliche Laden geht dann sehr einfach:

```
oshell newbcm bcm121
shutdown
```

Unter Linux ist das Update einfacher, hier kann während die Box läuft mit OSHELL eine Shell gestartet werden und von der aus können dann die diversen Dateien verändert werden. Es ist dabei sinnvoll, dass neben der OpenBCM-Mailbox auf demselben Rechner noch irgendeine andere Software läuft, die über Funk erreichbar ist, denn: Wenn beim Update etwas schief geht kann man sonst nicht mehr auf den Linux-Rechner zugreifen!

5. Konfiguration für den ersten Start

Nun steht die richtige Betriebsystem- und Hardwareumgebung zur Verfügung, es kann also mit der Konfiguration der Mailbox begonnen werden (für Tests reicht irgendeine Umgebung, am Anfang auch ohne Funkanschluss).

Was ist zu Konfigurieren?

- Die Grundeinstellungen (*init.bcm* und evtl. *init.l2*)
- Die Bulletin-Rubriken und Beschreibungen (*bulletin.bcm* und *boardinf.bcm*)
- Das Sysop-Passwort (*passwd.bcm*)
- Die Zeitsteuerung (*crontab.bcm*)
- Der Store&Forward (*fwd.bcm*)
- Definition des Autosysop (*asysop.bcm*)
- Sprachdefinitionsdatei (*speech.bcm*)
- Bakeneinstellungen (*beacon.bcm* und *beachead.bcm*)
- Reject und Hold Definitionen (*reject.bcm*)
- automatische Rubrikkonvertierungen (*convert.bcm*)
- automatische Verteilerkonvertierungen (*convat.bcm*)
- Runutil-Definitionen (*runutil.bcm*)
- Remotehost (*rhosts.bcm*)

5.1. Grundeinstellungen (*init.bcm* und evtl. *init.l2*)

Die Grundeinstellungen werden in der Datei *init.bcm* gespeichert. Diese Datei wird von der Box spätestens nach der Eingabe der hierarchischen Box-Adresse automatisch erstmalig angelegt und bei Änderungen immer automatisch aktualisiert. Folgende Vorgehensweise wird für den ersten Start empfohlen:

- Unter Linux und Windows ist es notwendig bereits vor dem ersten Start die Datei *rhosts.bcm* zu editieren
- Mailbox starten
- Unter Linux/Windows Login über Telnet ("telnet localhost 4719") bzw. unter DOS an der Konsole durch die Eingabe des eigenen Rufzeichens und Enter
- Den Befehl "BOXADDRESS db0aab.#bay.deu.eu" eingeben (oder besser die eigene Boxadresse)
- Durch den Befehl "MYCALL db0aab-8 db0aab-7" das Login/Forward-Rufzeichen der Mailbox einstellen

- Mit dem Befehl "MKBOARD / TMP" wird die Datei *bulletin.bcm* mit dem Defaulteintrag versehen (nur wenn noch kein *bulletin.bcm* vorhanden)
- Die Mailbox mit ENABLE in den aktiven Zustand versetzen
- Mailbox mit dem Befehl SHUTDOWN wieder beenden

Nun ist die Datei *init.bcm* angelegt, die eigene hierarchische Adresse eingegeben und das Boxrufzeichen eingestellt. Damit sind die wichtigsten Einstellungen getroffen. Folgende Parameter sollten in der Datei *init.bcm* noch überprüft werden:

boxheader
"Werbezeile" - Locator, Call des Sysop, Ort der BBS, z.B. "boxheader JN78ov Zwettl OP:OE3DJB"

sysopcall
Rufzeichen des Sysop (für Login an der Konsole)

infopath
Pfad zu den Bulletins, eventuell auf eine eigene Partition einstellen

userpath
Pfad zu den User-Mails, eventuell auf eine eigene Partition einstellen

userlife
Lifetime von Usermails, bei Bedarf erhöhen

dosinput
Nur bei DOS; je nachdem ob die Mailbox mit shroom geladen wurde oder nicht (Mit shroom: 0, ohne shroom: 1)

userpw
Passworteingabe durch User erlauben (userpw 1) oder abstellen (userpw 0)

Unter WinNT und Linux sollte außerdem noch die Datei *init.l2* angepasst werden, damit eine Anbindung an die Außenwelt stattfinden kann.

Die Dateien *init.bcm* und *init.l2* sind ausführlich in den Kapiteln "23.4. Initialisierungsdatei *init.bcm*" und "23.5. Initialisierungsdatei *init.l2*" dargestellt.

5.2. Bulletinrubriken (*bulletin.bcm* und *boardinf.bcm*)

Im Gegensatz zur FBB oder MSYS werden bei der OpenBCM-Mailbox bestimmte Bulletinrubriken definiert. Mails, die nicht in dieses Rubrikenschema passen, werden temporär angelegt. Lokal werden nicht in dieses Schema passende Rubriken nur dann angelegt, wenn der Parameter CREATEBOARD auf 1 gestellt wurde.

Nachfolgend ein kurzes Beispiel für eine *bulletin.bcm* im Amateurfunk:

```
; OpenBCM-Mailbox Bulletin Listfile
; Syntax: Rubrik Max.Lifetime [Min.Lifetime]
aktuell   120  30
 lokales  999 999
 termine  360  10
alle      120  10
 all       30   5
 tous      60   5
bus       360  90
 amtor    360 360
 aplink   360   0
 atv      360   0
```

```
 clover   360   0
 cw       180   0
 fax      360   0
 gtor     360   0
 pactor   360   0
 rtty     360   0
 sattv     90  90
 sstv     360 360
[...]
wetter     10   5
 meteo     10   5
 wx-info  365 365
z           5   5
```

In *bulletin.bcm* wird nicht nur die Struktur der Bulletinnamen angegeben, sondern auch die Default-, Maximal- und Minimal-Lebensdauer von Mails in der jeweiligen Rubrik.

Außer den im obigen Beispiel angegebenen, gibt es natürlich noch eine Vielzahl mehr Rubriken. Manche von diesen Rubriken sind Synonyme (6m, six, 50mc, 50mhz), solche Rubriken lassen sich bequem durch *convert.bcm* zusammenfassen (siehe hierzu Kapitel "5.11. Automatische Rubrikkonvertierung (*convert.bcm*)"). Andere wiederum sind am besten in der Rubrik TMP aufgehoben.

In der Datei *boardinf.bcm* können optional Rubrikbeschreibungen erstellt werden, die dann bei einem DIR oder LIST mit angezeigt werden. Das Format ist hierbei zeilenweise:

BOARDNAME Beschreibung

Ist für eine Rubrik in *boardinf.bcm* keine Beschreibung vorhanden, so wird diese logischerweise auch nicht angezeigt.

Nachfolgend ein kurzes Beispiel für eine *boardinf.bcm* Datei:

```
AKTUELL   Thema Aktuelles allgemein
termine   Terminkalender, Veranstaltungshinweise
ALLE      Mitteilungen an alle (deutschsprachig)
all       Mitteilungen an alle (englischsprachig)
cept      Infos Europaeische Fernmeldeverwaltungen
[...]
WETTER    Thema Wetterberichte allgemein
wefax     Wetterfaxbilder
```

5.3. Sysop-Passwort (*passwd.bcm*)

Ein Passwort für den Sysop kann in der Datei *passwd.bcm* eingegeben werden. Ohne dieses Passwort ist keine Remote-Sysoptätigkeit möglich. In der ersten Zeile dieser Datei steht das Passwort für die Sysop-Identifikation mittels BayCom-, in der zweiten das für MD2-, und in der dritten das für MD5-Passwort-Verfahren.

5.4. Zeitsteuerung (*crontab.bcm*)

Die Box führt zu festgelegten Zeiten verschiedene Prozesse aus. Der Startzeitpunkt dieser Prozesse wird in einer eigenen Datei, *crontab.bcm*, festgelegt. Die Zeitsteuerung ist eng mit der *crontab* von Unix verwandt.

In der Datei *crontab.bcm* wird die Zeitsteuerung der Mailbox festgelegt. Eine einfache *crontab.bcm* wird automatisch nach dem ersten Start der Mailbox erzeugt. Zusätzlich sollten insbesondere Routinen zur Datensicherung (Backup) hinzugefügt werden. Eine Ausführliche Beschreibung dieser Datei findet man im Kapitel "15. Zeitgesteuerte Prozesse".

5.5. Der Store & Forward (*fwd.bcm*)

ACHTUNG, WICHTIG: Dieses Kapitel bitte genau lesen! Die richtige Konfiguration der automatischen Weiterleitung von Mails ist eine der wichtigsten Aufgaben eines Boxsysops!

In der Datei *fwd.bcm* wird festgelegt, wie Bulletins und User-Mails weitergeleitet werden. Leider wird diese Weiterleitung sehr oft falsch konfiguriert. Die Folgen sind liegen bleibende Nachrichten oder eine unnötige Hin- und Herleitung von Nachrichten.

In den späteren Abschnitten dieser Beschreibung wird genau erklärt, wie der Forward aufgebaut wird, hier soll mit einigen Beispielen ein richtiger Forward demonstriert werden.

Also, zuerst einfache Beispiele: DB0BCM ist Deine neue Mailbox, ein einfacher Forward von Bulletins soll getestet werden. Der Sysop von DB0AAB ist bereit, seine Box als Forwardpartner zur Verfügung zu stellen. Es sollen nur Nachrichten mit dem Verteiler @DL ausgetauscht werden, also z.B. MEINUNG@DL, IBM@DL, SATTV@DL etc.

Die Forward-Datei *fwd.bcm* von DB0BCM sieht nun folgendermaßen aus:

```
DB0AAB - DB0AAB-8
 DL
```

Bei DB0AAB ist der folgende Eintrag für die Versuchsbox vorhanden:

```
[...]
DB0BCM - DB0BCM-8 DB0AAB
 DL
[...]
```

In der zweiten Zeile stehen um mindestens 1 Zeichen eingerückt (also ein Leerzeichen am Anfang der Zeile) jene Adressen, die weitergeleitet werden sollen, hier ist jeweils ein Eintrag für DL, also "...@DL" vorhanden.

In der ersten Zeile steht der Name der Zielbox (DB0AAB), die Zeiten zu denen das Forward ablaufen soll (im Normalfall ein "-" oder 24 "A") und Informationen dazu, wie die Verbindung zum Forward-Partner aufgebaut werden soll. Die Zielbox heißt "DB0AAB", es wird also kein SSID angegeben. Da die Box DB0BCM am Userzugang von DB0AAB (Digi) angeschlossen ist, kann der Forward-Partner direkt connected werden, dazu wird im Feld für den Verbindungsaufbau einfach "DB0AAB-8" angegeben. In die umgekehrte Richtung, also von DB0AAB-8 zu DB0BCM-8 muss der Connect "via" DB0AAB erfolgen, weil die Mailbox DB0AAB-8 nicht direkten Zugriff auf den Benutzerzugang hat, sondern über Ethernet am Knoten DB0AAB angeschlossen ist. Der Eintrag bei DB0AAB-8 lautet deshalb "DB0BCM-8 DB0AAB".

Verwirrend? Nicht so schlimm, Du kannst von anderen OpenBCM-Mailboxen deren *fwd.bcm* mit dem Befehl "DIR PATH" abrufen, und mit anderen Sysops über die richtige Einstellung des Connect-Weges diskutieren.

Aber jetzt geht's weiter:

Bisher wurden nur Mails mit dem Verteiler "@DL" ausgetauscht, jetzt sollen noch zusätzlich die Verteiler "@BAYCOM" (zur BCM-Softwareverteilung) und "@EU" ausgetauscht werden.

Bei DB0BCM lautet der Eintrag nun:

```
DB0AAB - DB0AAB-8
 BAYCOM DL EU
```

Und bei DB0AAB:

```
[...]
DB0BCM - DB0BCM-8 DB0AAB
 BAYCOM DL EU
[...]
```

Geändert hat sich nicht viel, es sind jeweils nur Einträge für die Verteiler BAYCOM und EU hinzugekommen.

Bisher machte es überhaupt nichts, wenn mal die Box DB0BCM nicht funktionierte - schließlich waren ja nur Bulletins gespeichert, die auch in DB0AAB vorhanden sind. Sie wurden beim Weiterleiten kopiert (und nicht in der Absenderbox nach der Weiterleitung gelöscht). Bei Usermails übernimmt der Sysop jedoch eine gewisse Verantwortung: Wenn die Box Usermails empfängt, so werden sie bei der Box, die diese Mails übertragen hat, nach der erfolgreichen Weiterleitung gelöscht. Ist die eigene Box falsch konfiguriert, so sind diese Mails unwiederbringlich verloren. Deshalb: Erst dann Usermails im Forward empfangen, wenn Du glaubst Dich mit dem Forward gut auszukennen.

Du hast Dich deshalb vorerst mal dazu entschieden, von Deiner Box keine Nachrichten für andere Boxen zu empfangen, es sollen lediglich alle Nachrichten an DB0AAB weitergeleitet werden, die nicht für die eigene Box bestimmt sind, und von DB0AAB sollten nur Nachrichten für DB0BCM empfangen werden. DB0BCM liegt genauso wie DB0AAB in Bayern, die hierarchischen Adressen lauten "DB0AAB.#BAY.DEU.EU" bzw. "DB0BCM.#BAY.DEU.EU". Damit braucht DB0AAB lediglich einen einfachen Eintrag in *fwd.bcm* zu machen:

```
DB0BCM - DB0BCM-8 DB0AAB
 DB0BCM BAYCOM DL EU
```

Bei DB0BCM-8 müssen alle sinnvoll adressierten User-Mails nach DB0AAB geschickt werden. Deshalb müssen alle Kontinente (außer Europa), alle europäischen Staaten (außer Deutschland), alle deutschen Bundesländer (außer Bayern) und alle bayerischen Boxen (außer der eigenen) beim *fwd.bcm*-Eintrag für DB0AAB angegeben werden.

Umständlich? Nur auf den ersten Blick. Denn falls das Forwarding (später) auf mehrere Nachbarboxen aufgeteilt wird, so brauchen die Einträge nur von einer Box zur anderen Box verschoben werden.

Die Datei *fwd.bcm* sieht bei DB0BCM nun folgendermaßen aus:

```
DB0AAB - DB0AAB-8
; Verteiler für Bulletin-Mails
 BAYCOM DL EU
; Usermails: Kontinente (außer Europa)
 .AF .AFRC .AS .ASIA .AU .AUST .CEAM .CARB .MDLE .NA .NOAM
 .OC .OCEA .SA .SOAM
; Staaten (außer Deutschland)
 .AUT .BEL .BGR .BIH .CHE .CZE
 .DNK .ESP .EST .FIN .FRA .GBR .GIB
 .GRC .HRV .HUN .IRL .ITA .LTU .LUX
 .LVA .MKD .MLT .NLD .NOR .POL .PRT
 .ROM .RUS .SVK .SVN .SWE .TUR
 .UKR .YUG
 .HUN.EURO .ITA.EURO
; Deutsche Bundesländer (außer Bayern)
 .#NRW .#BLN   .#RPL  .#BRB   .#SAA
 .#BW  .#SAR   .#HB   .#SAX   .#HES   .#SLH
 .#HH  .#THR   .#MVP  .#NDS
; Bayerische Boxen (außer eigene Box)
 DB0AAB DB0ABH DB0AHO DB0AST DB0BOX DB0FFB DB0FP
 DB0FSG DB0GAP DB0IGL DB0IRS DB0ISW DB0KCP DB0KFB
 DB0KP  DB0LAN DB0MAK DB0MAS DB0MFG DB0MWS DB0PV
 DB0RGB DB0SL  DB0ULM DB0WGS DB0ZB  DB0ZKA DF0ANN
 DK0MUC DK0MUN
```

Die Namen der Staaten sind durch die ISO-3166 genormt, die Kontinente sind nicht genormt, die hier angeführte Liste entspricht der Erfahrung des Autors.

DB0BCM war nun einige Wochen erfolgreich im Betrieb, die Box wird nun nicht mehr am Userzugang von DB0AAB, sondern über einen eigenen Link an DB0AAB angebunden. Die Box wird an einen RMNC-Digi angeschlossen. Zusätzlich sollen jetzt auch die "MyBBS-Infos" zwischen DB0AAB und DB0BCM ausgetauscht werden. Dies geschieht mit dem Eintrag "$WP". Die neuen Forward-Dateien sehen nun so aus:

Bei DB0AAB:
```
[...]
DB0BCM - DB0BCM-8 DB0AAB
 DB0BCM BAYCOM DL EU $WP
[...]
```

Bei DB0BCM:
```
DB0AAB - DB0AAB-8 DB0BCM
; Verteiler für Bulletin-Mails
 BAYCOM DL EU $WP
; Usermails: Kontinente (außer Europa)
 .AF .AFRC .AS .ASIA .AU .AUST .CEAM .CARB .MDLE .NA .NOAM
 .OC .OCEA .SA .SOAM
; Staaten (außer Deutschland)
 .AUT .BEL .BGR .BIH .CHE .CZE
 .DNK .ESP .EST .FIN .FRA .GBR .GIB
 .GRC .HRV .HUN .IRL .ITA .LTU .LUX
 .LVA .MKD .MLT .NLD .NOR .POL .PRT
 .ROM .RUS .SVK .SVN .SWE .TUR
 .UKR .YUG
 .HUN.EURO .ITA.EURO
; Deutsche Bundesländer (außer Bayern)
 .#NRW .#BLN   .#RPL  .#BRB   .#SAA
 .#BW  .#SAR   .#HB   .#SAX   .#HES   .#SLH
 .#HH  .#THR   .#MVP  .#NDS
; Bayerische Boxen (außer eigene Box)
 DB0AAB DB0ABH DB0AHO DB0AST DB0BOX DB0FFB DB0FP
 DB0FSG DB0GAP DB0IGL DB0IRS DB0ISW DB0KCP DB0KFB
 DB0KP  DB0LAN DB0MAK DB0MAS DB0MFG DB0MWS DB0PV
 DB0RGB DB0SL  DB0ULM DB0WGS DB0ZB  DB0ZKA DF0ANN
 DK0MUC DK0MUN
```

Der Eintrag "$WP" ist hinzugekommen, und beim Connect-Weg kam ein "via DB0BCM" hinzu, denn die DB0BCM-Box muss nun über den RMNC-Digi das Ziel connecten.

So, jetzt ist es fast geschafft, nun wird zusätzlich noch ein Store&Forward mit OE5BCM betrieben. Dorthin sollen alle Bulletins und alle Usermails für Österreich - außer für Tirol - geschickt werden. Zwischen DB0AAB und DB0BCM sollen nun auch die Verteiler "@WW" und "@EU" ausgetauscht werden. OE5BCM schickt nur Mails für DB0BCM.

Zusätzlich soll der Verteiler THEBOX hinzukommen. Mit diesem Verteiler werden eigentlich fast nur Mails aus der "F"-Rubrik verteilt. Diese Rubrik ist - wie auch alle anderen einbuchstabigen Rubriken normalerweise nur nach Eingabe des Sysoppasswortes les- und beschreibbar. Die Rubrik F dient zum Austausch von allgemeinen Sysopinformationen, früher wurden mit diesem Verteiler auch White-Page-Informationen in einem älteren Format ausgetauscht.

fwd.bcm bei DB0AAB:
```
[...]
DB0BCM - DB0BCM-8 DB0AAB
 DB0BCM BAYCOM THEBOX DL EU WW $WP
[...]
```

fwd.bcm bei OE5BCM:
```
[...]
DB0BCM - DB0BCM-8 OE5BCM
 DB0BCM BAYCOM THEBOX DL EU WW $WP
[...]
```

fwd.bcm bei DB0BCM:
```
DB0AAB - DB0AAB-8 DB0BCM
; Verteiler für Bulletin-Mails
 BAYCOM THEBOX WW EU DL $WP
; Usermails: Kontinente (außer Europa)
 .AF .AFRC .AS .ASIA .AU .AUST .CEAM .CARB .MDLE .NA .NOAM
 .OC .OCEA .SA .SOAM
; Staaten (außer Deutschland/Österreich)
 .BEL .BGR .BIH .CHE .CZE
 .DNK .ESP .EST .FIN .FRA .GBR .GIB
 .GRC .HRV .HUN .IRL .ITA .LTU .LUX
```

```
 .LVA .MKD .MLT .NLD .NOR .POL .PRT
 .ROM .RUS .SVK .SVN .SWE .TUR
 .UKR .YUG
 .HUN.EURO .ITA.EURO .ROM.EURO
; Tiroler Mailboxen
 .#OE7.AUT
; Deutsche Bundesländer (außer Bayern)
 .#NRW .#BLN   .#RPL  .#BRB   .#SAA
 .#BW  .#SAR   .#HB   .#SAX   .#HES   .#SLH
 .#HH  .#THR   .#MVP  .#NDS
; Bayerische Boxen (außer eigene Box)
 DB0AAB DB0ABH DB0AHO DB0AST DB0BOX DB0FFB DB0FP
 DB0FSG DB0GAP DB0IGL DB0IRS DB0ISW DB0KCP DB0KFB
 DB0KP  DB0LAN DB0MAK DB0MAS DB0MFG DB0MWS DB0PV
 DB0RGB DB0SL  DB0ULM DB0WGS DB0ZB  DB0ZKA DF0ANN
 DK0MUC DK0MUN
OE5BCM - OE5BCM-8 DB0BCM
 BAYCOM THEBOX WW EU DL
 .AUT
```

Beim Forwardeintrag für DB0AAB ist ".AUT" durch ".#OE7.AUT" ersetzt worden.

Ob das alles stimmt? Du wirst es sehen. Mails die aufgrund ihrer Adresse nicht weitergeleitet werden können, werden in der Datei *trace/unknown.bcm* vermerkt. Dort landen neben allen sinnlos und falsch adressierten Mails auch jene Mails, die wegen eines Fehlers in der Forward-Datei nicht weitergeleitet werden konnten. Mit dem Befehl UNKNOWN wird der Inhalt dieser Datei ausgegeben.

Im Kapitel "9. Forwarding komplett einrichten" wird die Forward-Konfiguration nochmals genauer erklärt.

5.6. Autosysop (*asysop.bcm*)

Die Datei *asysop.bcm* ermöglicht die Einrichtung einer automatischen Sysopberechtigung bei bestimmten Links, sofort nach dem Login. Eine Passworteingabe ist dann nicht notwendig.

Wichtig: Die Einstellungen sollten gut durchdacht werden, da sich hier sonst ein Sicherheitsrisiko auftun kann! Ist die Datei *asysop.bcm* nicht vorhanden, so wird diese Funktion auch nicht aktiviert. Sicherheitsfanatiker verzichten also auf diese Datei!

Format von *asysop.bcm*:

```
Rufzeichen uplink   downlink
```

Beispiel:

```
DJJ812 none     none            (Direkteinstieg)
DJJ812 DBO812-2 DBO812-2        (wenn das SSID des Ports, über den man einsteigt, 2 ist)
```

Soll der Up- oder Downlink nichts enthalten, wird "none" angegeben. Der Downlink ist das Rufzeichen des Node bzw. Digipeater, der der Mailbox am nächsten liegt. Der Uplink ist das Rufzeichen des Nodes, der am weitesten von der Mailbox entfernt ist, bzw. über den der Benutzer ins Netz einsteigt. Jede Zeile muss mit einem Return abgeschlossen werden.

Achtung:
Der Uplinkdigi ist leicht fälschbar, der Downlinkdigi theoretisch nie, da es sich meist um den eigenen handelt. Allerdings sollte man darauf achten, dass sich dieser nicht auch durch Eingriffe von Außen verstellen lässt. Bei Einstieg über den eigenen Node entspricht der Uplink dem Downlink, hier auf SSID achten (bei Flexdigi 3.3e von außen leicht änderbar, ab Flexdigi 3.3g sicherer, da es hier ein neues Passwortverfahren gibt).

- Im Uplink befinden sich bei IP-Logins die Schlüsselwörter HTTP, TCP/IP etc., sie können ebenfalls verwendet werden.

- Wird bei beiden "none" angegeben, handelt es sich um einen Direktconnect ohne Node oder Digipeater dazwischen.
- Besitzt der Digi das SSID 0 oder kein SSID, gilt das Rufzeichen ohne SSID.

5.7. Remotehost (*rhosts.bcm*)

Die Datei *rhosts.bcm* wurde bereits in Kapitel "4. Installation" kurz beschrieben. In ihr wird definiert, welche IP-Nummern/Hosts vertrauenswürdig sind, so dass beim Login über einen solchen Host nur das Rufzeichen verlangt wird und der eingeloggte Benutzer danach automatisch Sysop in der Box ist.

Ist der Uplink-Rechner nicht in *rhosts.bcm* enthalten, so ist zum Loginrufzeichen noch das TTY-Passwort (TTYPW) anzugeben und der Benutzer erhält keinen Sysopstatus. Es können also nur Benutzer in die Box, denen zuvor von einem Sysop ein TTY-Passwort zugeteilt worden ist.

Hinweis: Auch die eigene IP-Adresse und das Symbol "localhost" muss in *rhosts.bcm* enthalten sein, wenn auf dem lokalen Rechner ein Login ohne Passwort möglich sein soll. Für den ersten Login ist es also in jedem Fall erforderlich die Datei *rhosts.bcm* mit der eigenen IP-Adresse bzw. dem Hostnamen anzulegen.

Achtung: Der eigene Rechner sollte nur dann in *rhosts.bcm* eingetragen werden, wenn sichergestellt ist, dass nur Sysops darauf Zugriff haben. Wenn etwa WAMPES oder die AX.25-Utils mit AXSPAWN installiert sind, so kann jeder auf den Rechner zugreifen, hier sollte auch der eigene Boxrechner nicht im *rhosts.bcm* enthalten sein.

5.8. Sprachdefinitionsdatei (*speech.bcm*)

In der Datei *speech.bcm* wird die Zuordnung der vorhandenen Sprachdateien (*msg/messages.**) zu den Rufzeichenmuster festgelegt, die diese Sprache als Defaultwert beim ersten Login erhalten sollen.

Es können beliebig viele Rufzeichenmuster zu einem Sprachkenner zugeordnet werden, müssen aber jeweils durch Leerzeichen getrennt sein und die Zeile darf insgesamt 80 Zeichen nicht überschreiten.

Das Format ist zeilenweise also wie folgt:

MSG-Kenner Rufzeichenmuster

Beispiel:

DL DG DH DL DO OE

Hier werden der Sprache "DL" die Rufzeichenmuster "DG", "DH", "DL", "DO" und "OE" zugeordnet.

Die Zuordnungen kann man sich in der Mailbox mit "a s" ausgeben, sie werden in der Auflistung dort in den Klammern angezeigt.

Eine Besonderheit ist, wenn als Rufzeichenmuster eine Sprache, also z.B. "DL", in Klammern angegeben wird. Dann wird für HELP-, CTEXT-, QTEXT-, etc. -Dateien diese Sprache, für die reinen Meldungen jedoch die "messages"-Datei der eingestellten Sprache verwendet.

Wird gar kein Rufzeichenmuster angegeben, so ist für diese Dateien die Sprache "GB" als Standard eingestellt.

Die erste Zeile der Dateien *msg/messages.** enthält übrigens eine Beschreibung mit Versionsnummer der Sprachdatei, die bei "a s" ebenfalls angezeigt wird.

Die Datei *speech.bcm* wird, wenn die Datei nicht vorhanden ist, abhängig vom Parameter CALLFORMAT in *init.bcm* entweder für Amateur- oder CB-Funk automatisch erzeugt.

5.9. Bakeneinstellungen (*beacon.bcm* und *beachead.bcm*)

In der Datei *beacon.bcm* wird das Sendeziel für die Bakenausstrahlung definiert.

Format für DOS:

<mycall> <tocall> [<viadigi> ...]
<mycall> <tocall> [<viadigi> ...]

Format für Linux/Windows:

<tocall> [<viadigi> ...]
<tocall> [<viadigi> ...]

Die Parameter haben folgende Bedeutungen:

<mycall> Rufzeichen, unter dem die Bake gesendet werden soll
<tocall> Zieladresse (UI-Adresse) der Bake
<viadigi> evtl. die Digis, über den die Bake gesendet werden soll

Beispiel für DOS, unter Linux und Windows muss die MyCall-Angabe "DB0AAB-8" weggelassen werden:

DB0AAB-8 MAILS DB0AAB Bake wird auf DB0AAB abgestrahlt
DB0AAB-8 MAILS DB0AAB OE7XAR Bake wird auf OE7XAR abgestrahlt

Soll am Anfang jeder Bakenausstrahlung ein fixer Text ausgesendet werden, so ist dieser in der Datei *beachead.bcm* abzuspeichern. Angegebene Makros werden dabei expandiert.

Das Kapitel "17. Mail-Bake" befasst sich noch einmal näher mit der Mail-Bake.

5.10. Reject und Hold (*reject.bcm*)

In der Datei *reject.bcm* können Regeln definiert werden, wie mit empfangenen Mails umgegangen werden soll. Die Mails können dabei abgelehnt (rejected) oder für den Forward zeitlich gesperrt (auf Hold gesetzt), d.h. so lange in der Mailbox nur lokal gespeichert werden, bis der Sysop sie für den Forward freigibt, werden. Kommentare können in der Datei beliebig beigefügt werden, müssen aber immer mit einem ";" beginnen.

Das genaue Format einer Reject-Zeile in *reject.bcm* ist:

<Aktion> [<sender] [>dest] [$bid] [@at] [.B | .P] ;Kommentar

Bedeutung von:

- <Aktion>:
  R = Nachricht wird abgewiesen (rejected)
  G = Nachricht wird abgewiesen (rejected), wenn Lokal ohne AX25-PW eingegeben
  E = Nachricht wird abgewiesen (rejected), wenn Lokal ohne AX25/TTYPW eingegeben
  F = Nachricht wird auf Hold gesetzt, wenn S&F nicht passwortgeschützt ist
  H = Nachricht wird auf Hold gesetzt
  L = Nachricht wird auf Hold gesetzt, wenn Lokal eingegeben
  P = Nachricht wird auf Hold gesetzt, wenn ohne AX25-PW eingegeben
  O = Nachricht wird auf Hold gesetzt, wenn ohne AX25/TTYPW eingegeben
- [<sender] Absenderrufzeichen
- [>dest] Empfänger (Rubrik oder Rufzeichen)

- [$bid]   Ein bestimmtes (Teil-)BID
- [@at]   Adresse (Verteiler oder Mailboxrufzeichen)
- .B   Bulletin
- .P   persönliche Nachricht

Die Aktionen E und O sind nur in der Linux/Windows Version möglich!

Als Jokerzeichen sind "?" (beliebiges Zeichen) und "*" (mehrere beliebige Zeichen oder kein Zeichen) möglich. Ein "*" kann auch am Anfang oder mitten im String angegeben werden. Als String kann ein "regulärer Ausdruck" verwendet werden, siehe dazu auch Kapitel "28.6. Reguläre Ausdrücke".

Zur Negierung eines Ausdrucks kann dem String ein "!" vorangestellt werden, z.B. würde die Zeile

```
    R @DB0IRS >!DH3MB
```

bewirken, dass alle Nachrichten rejected werden, die an DB0IRS adressiert sind und nicht an DH3MB gehen (nicht sinnvoll).

Ferner kann eine ODER-Verknüpfung genutzt werden, so würde eine Zeile

```
    G .B <!(DH8YMB,DL9CU
```

bewirken, dass alle Rubrikmails abgelehnt werden, die - mit Ausnahme von DH8YMB und DL9CU - ohne passwortgeschützten Login eingespielt werden. Eine ODER-Verknüpfung ohne Negierung ist auch möglich, macht in der Regel aber wenig Sinn.

Mit dem *reject.bcm*-Feature sollte vorsichtig umgegangen werden, eine Zeile

```
    R <*CB*
```

würde mehr sperren als man auf dem ersten Blick vermutet und würde manche Leute nicht sehr erfreuen!

Weitere Beispiele:

```
; CB-Rufzeichen sperren
  R <CB?CB
; FLOHMARKT-Mails
  R >FLOHM*
; Weltweite MEINUNGs-Mails gar nicht erst annehmen
  R >MEINU* @WW
; df0ar darf nur Mails an OE3dzw senden
  R <^DF0AR$ >!^OE3DZW$ ...
; Einige BIDs von CB-Mailboxen sperren
  R $*DBO123*
  R $*NL3DGH*
; Generell alle Rubrikmails, die ohne AX25-Passwort eingespielt wurden
; auf "HOLD" setzen (Parameter HOLDTIME muss gesetzt sein!)
  P .B
; alle ohne AX25-Passwort lokal eingegebene Rubrikmails ablehnen
  G .B
; alle ohne AX25-Passwort lokal eingegebene Rubrikmails ablehnen, außer
; wenn sie von den Calls, die mit DB0 beginnen, oder dem Call DH8YMB stammen
  G .B <!(^DB0,^DH8YMB
```

Eine Zeile kann mehrere Bedingungen enthalten. Nur wenn alle Bedingungen erfüllt sind, wird die <Aktion> ausgeführt.

Neben dem reinen Editieren der Datei *reject.bcm* kann auch mit dem internen REJECTEDITOR gearbeitet werden, sofern dieser in der Mailbox einkompiliert wurde (unter DOS aus Speichermangel evtl. nicht).

Mit HOLDTIME kann der Sysop die Haltezeit für Nachrichten einstellen, die ein Hold-Kriterium erfüllen. Ein sinnvoller Wert für HOLDTIME ist meist "48" (entspricht 2 Tage). Mit DIR HOLD lassen sich alle angehaltenen Nachrichten aufzeigen. Auf HOLD gesetzte Mails kann der Sysop nach Durchsicht mit dem Befehl "hold -u <rubrik> <nr>" oder "f -h <rubrik> <nr> @ <neuer verteiler>" wieder in den Forward geben.

Im Kapitel "12. PW & Hold/Reject Funktionen der Mailbox" wird noch einmal auf die Möglichkeiten von *reject.bcm* eingegangen.

5.11. Automatische Rubrikkonvertierung (*convert.bcm*)

Die Datei *convert.bcm* dient der automatischen Transferierung von Mails in Abhängigkeit von den angegebenen Bedingungen. Auch die maximale Lifetime kann in Abhängigkeit von Bedingungen einzeln geändert werden. In früheren Mailboxversionen wurden hierfür die Dateien *convname.bcm* und *convlife.bcm* benutzt. Diese beiden Dateien werden seit der Einführung von *convert.bcm* jedoch nicht mehr benötigt und können gefahrlos gelöscht werden.

Bedeutung der Einträge:

```
   ;              Kommentarzeile
```

Eine Zeile ohne Kommentar darf maximal 54 Zeichen lang sein, mit angefügtem Kommentar 80 Zeichen inklusive CR/LF.

Zeichen mit besonderer Bedeutung: < @ ~ $ % &

```
   <Rufzeichen    Absender
   @Adresse       Box oder Region
   ~Wort          sucht ganze Worte im Titel
   text           sucht "text" im Titel
```

Die Zeichen % $ & sind nur ohne Argument zu verwenden.

```
   $              per S&F von anderer Box empfangen
   %              Benutzer Eingabe
   &              Benutzer Eingabe ohne Passwort
```

Wichtig: als Trennzeichen nur Leerzeichen verwenden, keine TABs. Es können mehrere Bedingungen angegeben werden, die "und" verknüpft werden. Um eine "oder" Bedingung zu erreichen, benutzt man eine weitere Zeile. Es müssen immer alle Bedingungen erfüllt sein. Groß-/Kleinschreibung ist irrelevant.

Ein Wort ist von Leerzeichen umgeben, also bedeutet ~NO = " NO ", oder wenn es am Titelanfang steht: "NO ", oder wenn es am Titelende steht " NO".

LT 000 bedeutet, dass die maximale Lifetime aus *bulletin.bcm* genommen wird. Die maximale Lifetime aus *bulletin.bcm* kann mit *convert.bcm* nach oben oder unten geändert werden. Eine LT 0 kann man mit *convert.bcm* nicht erreichen, LT 1 sollte reichen, aber nur, wenn die minimale Lifetime aus *bulletin.bcm* nicht unterschritten wird. Die Reihenfolge der Einträge ist zu beachten, weil die erste Übereinstimmung verwendet wird.

Bei <, @, und text kann man "reguläre Ausdrücke" anwenden (siehe hierzu auch Kapitel "28.6. Reguläre Ausdrücke"), nicht jedoch bei ~.

Beispiel für eine *convert.bcm*:

```
;Example convert.bcm
;from    to        LT mode/scan title   ; comment also possible here
;                                                   <max. line length
BAYBOX   BAYBOX   000 $                 ; do not convert if received via S&F
F        F        000 @THEBOX           ; do not convert if address is THEBOX
*        Z        002 <DL3XYZ           ; each bulletin from DL3XYZ to "Z"
DUTCH    RTTY     014 <PA3ESF           ; board DUTCH from PA3ESF to RTTY
*        Z        002 @CBCALL           ; invalid h-address to Z with LT 2
*        AMSAT    180 @AMSAT            ; all @AMSAT to AMSAT
;*        AMSAT    180 @ AMSAT           ; wrong, no blank after @, <, or ~
6M*      SIX      090 $                 ; $ = if received from a box
50*      SIX      090 $                 ; $ = if received from a box
TV*      SATTV    180 %                 ; % = if received from a user
*        H        002 &                 ; & = if from a user without PW
*        H        002 & @WW             ; &    "     and @WW
*        H        002 & @EU             ; &    "     and @EU
; title scan examples:
*        PROBLEM  014 ?                 ; all qüstions to PROBLEM with LT 14
*        CONTEST  014 ~CONTEST          ; ~ means word match
```

```
*        Z        002 ~NO TEXT          ; if word NO and "TEXT" matches
*        SUCHE    000 WANT              ; if "WANT" anywhere in title
*        BILDER   000 JPG               ; if JPG in title
DX*      QSLMGR   000 QSL
; regular expression examples (see also Help REGular_expressions):
*        Z        002 ^R:99             ; title starts with "R:99"
*        SUCHE    000 ^S:               ; title starts with "S:"
; 'and'-combination of blank separatet items:
*        SUCHE    000 ? QSL             ; if "?" and "QSL" in title
*        DIPLOME  360 $ @AMSAT ~AWARD DX ; all items must match
; only LT conversion:
KEPLER   *        014 $                 ; if received via S&F
;KEPLER   *        014                  ; wrong, that does not work
KEPLER   *        014 MIR DAY           ; title depending
; only to-field conversion:
CBMAIL   Z
ASCII    BILDER
JPG      BILDER
; wildcard for multi matching:
IMAGE*   BILDER
DX*      DXNEWS                         ; min. 2 chars before *
; end of example file
```

Jeder Benutzer der Mailbox kann sich die komplette *convert.bcm* durch Eingabe des Befehls "convert -a" anzeigen lassen. Wird zusätzlich ein <Suchtext> mit angegeben, werden alle Zeilen ausgegeben, die diesen Suchtext enthalten. Der <Suchtext> wird dabei als regulärer Ausdruck ausgewertet.

Zum bequemen Editieren der *convert.bcm* ist es möglich, den in die Mailbox eingebauten CONVEDITor zu benutzen, sofern dieser einkompiliert wurde (Option "DF3VI_CONV_EDIT" in *config.h*).

5.12. Automatische Verteilerkonvertierung (*convat.bcm*)

Die Datei *convat.bcm* enthält Konvertierungen für ungültige Verteiler. Die Konvertierung bezieht sich nur auf die lokal eingespielten Rubrikmails. Der Verteiler von im S&F empfangenen Nachrichten wird nicht verändert, jedoch wird eine Nachricht @oedl so weitergeleitet, wie @dl.

Ein Beispiel:

```
; Format: <alter Verteiler> <neuer Verteiler>
; jeweils ein Paar pro Zeile, durch Leerzeichen getrennt
OEDL DL
DLOE DL
OEBA DL
BAOE DL
ALLE DL
ALL  WW
EURO EU
EUR  EU
AUT  OE
DEU  DL
```

Gibt ein User "s meinung@oedl" ein, so wird die Mail an MEINUNG @DL weitergeleitet. Wird im S&F eine Mail an INFO @EUR" empfangen, so wird die Nachricht weitergeleitet, als wäre sie an @EU adressiert. Ein eigener Eintrag für EUR oder OEDL in der Datei *fwd.bcm* ist somit nicht notwendig, er wird ignoriert. Wichtig: Der Verteiler von im S&F empfangenen Mails wird nicht verändert, er wird nur intern anders behandelt, wenn für ihn ein Eintrag in dieser Datei vorhanden ist!

Mit dem Befehl CONVAT kann man sich die Datei *convat.bcm* ausgeben lassen.

5.13. Runutils (*runutil.bcm*)

Um ein Programm für Benutzer zugänglich zu machen, muss das Programm in einer Datei namens *runutil.bcm* eingetragen sein. Ansonsten ist die Abarbeitung des Programms vollkommen mit der beim Befehl OSHELL identisch.

Die Datei *runutil.bcm* hat z.B. folgendes Format (Linux-Beispiel):

```
;Befehl   Programm            Optionen Kommentar
BIN*READ  binread                      Gepacktes Auslesen von Mails
MEM*INFO  "cat /proc/meminfo" -qc      Anzeigen der Speicher-Auslastung
PSL*INUX  "ps aux"                     alle Linux-Prozesse anzeigen
FILENAME  filename            -qs      Dateiname einer Mail suchen  (SYSOP)
FM*AIL    fmail               -s       Erzeugen einer F-Mail        (SYSOP)
USERSTAT  runutils/userstat   -s       Userstatistik                (SYSOP)
READSTAT  runutils/readstat            Boardstatistik
FL*OG     runutils/flog                S&F-Statistik Digis (flog <yymmdd>)
UL*OG     runutils/ulog                Useraufkommen (ulog <yymmdd>)
QTH       qth                 -I       Locator abfragen/berechnen
NEWS      news                         News und Termine
SYSINFO   runutils/sysinfo             Systeminfos
7*MAIL    7mail               -faD     Als Mail per 7Plus lesen   (FILESURF)
7G*ET     7get                -faD     Auslesen per 7plus         (FILESURF)
7C*OR     7cor                -faD     COR-File auf ERR erstellen (FILESURF)
BS*GET    bsget               -faD     Auslesen in Binärteilen    (FILESURF)
V*IEW     fileview            -faD     Dateiarchive auflisten     (FILESURF)
I*NFO     fileinfo            -faD     Dateibeschreibung lesen    (FILESURF)
F*IND     filefind            -faD     Dateien suchen             (FILESURF)
N*EW      filenew             -faD     neue Dateien listen        (FILESURF)
XD*IR     filexdir            -faD     Extended "DIR"             (FILESURF)
FSE*DIT   fsedit              -fasD    Dateibeschreibung erstellen(FILESURF)
FSD*EL    fsdel               -fasD    Dateibeschreibung löschen  (FILESURF)
SYS7M*AIL sys7mail            -fasD    7Mail ohne Limits          (FILESURF)
HILFE     fshelp              -faD     zusaetzl. Hilfe für FS     (FILESURF)
```

Die Felder haben folgende Bedeutung:

- Befehl
  Das ist der Befehl in der Mailbox, mit dem das Run-Utility gestartet werden kann. Die Buchstaben vor dem Stern müssen eingegeben werden, damit der Befehl erkannt wird, die Buchstaben hinter dem Stern sind optional.
- Programm
  Das ist das externe Linux/DOS/Windows-Programm, das bei der Eingabe des Befehls aufgerufen wird. Enthält der Shell-Aufruf ein Leerzeichen, ist dieser Parameter in Anführungszeichen zu setzen.
- Optionen
  Dieses Feld ist optional. Es besteht aus einzelnen Buchstaben, denen ein "-" vorangestellt werden muss. Die Buchstaben beeinflussen den Aufruf des Run-Utilities und haben folgende Bedeutung:
  - -S: Befehl kann nur vom Sysop ausgeführt werden.
  - -F: Befehl kann nur im FileSurf-Modus ausgeführt werden.
  - -Q: Die Datei *rundat.bcm* wird nicht erzeugt.
  - -C: Die vom User eingegeben Parameter werden nicht an das externe Programm übergeben.
  - -I: Während das externe Programm läuft, werden Eingaben des Users nicht an dieses Programm durchgereicht, sondern nach Beendigung des Programms von der Box interpretiert (nur gültig für die Linux- und Windows-Version).
  - -P: Das Programm kann nur von Usern ausgeführt werden, die sich mit einem Passwort eingeloggt haben.
  - -D: Es werden wie bei der DPBox Umgebungsvariablen gesetzt, die den Angaben in *rundat.bcm* entsprechen (nur gültig für die Linux- und Windows-Version).
  - -T: Es werden auch die Zeichen ">", "<" und "|" an das externe Programm durchgereicht (nur gültig für die Linux-Version).
- Kommentar
  Eine kurze Beschreibung der Funktion des Run-Utilities (< 80 Zeichen).

Außerdem gelten folgende Regeln für diese Datei:

- Kommentare werden mit einem ";" eingeleitet.
- Die einzelnen Felder werden mit einem oder mehreren Blanks getrennt.

6. Funkanschluss

Auf einem Rechner kann je nach Betriebssystem neben der Mailbox auch noch andere Packet Radio-Software laufen. Außerdem können natürlich mehrere Rechnersysteme mit verschiedener Software laufen, die dann miteinander verbunden werden. Beispielhaft seien hier einige Kombinationen genannt:

- DB0AAB: An der FH-München laufen Digi und Mailbox jeweils auf einem eigenen Rechner. Die beiden Systeme sind via Ethernet miteinander verbunden. Der Knoten läuft unter DOS mit PC/Flexnet, der Mailboxrechner läuft unter Linux und dient gleichzeitig noch als TCP/IP-Server. Zusätzlich läuft noch eine Sprachmailbox, die ebenfalls am Ethernet-Bus hängt.
- OE1XLR/OE1XAB: Hier dient ein Rechner sowohl als Knoten als auch als Digi. Der Funkbetrieb erfolgt mit PC/Flexnet und USCC-Karten.
- OM0NVA: Bei diesem System ist die Mailbox über die serielle Schnittstelle des Rechners mit einem RMNC/Flexnet-Knoten verbunden.

Es sind natürlich noch viele weitere Konfigurationen denkbar. Im Folgenden soll aber hauptsächlich beschrieben werden, wie man die OpenBCM-Mailbox mit dem Digipeater verbindet.

Der Anschluss an die Außenwelt ist unter DOS mit PC/Flexnet möglich. Durch die Verwendung von Flennet gibt es nur einen "breiten" Datenkanal auf dem die Kommunikation zwischen Box und Flennet erfolgt. Damit gibt es nicht die vom Hostmode her bekannten Beschränkungen auf irgendwelche statischen TNC-Kanäle. Auf welchem Funkport die Ausgabe erfolgt, hat für die Mailbox keine Bedeutung.

Unter Linux erfolgt der Funkanschluss mit einem integrierten Layer2 oder mit dem Linux-Kernel-AX.25.

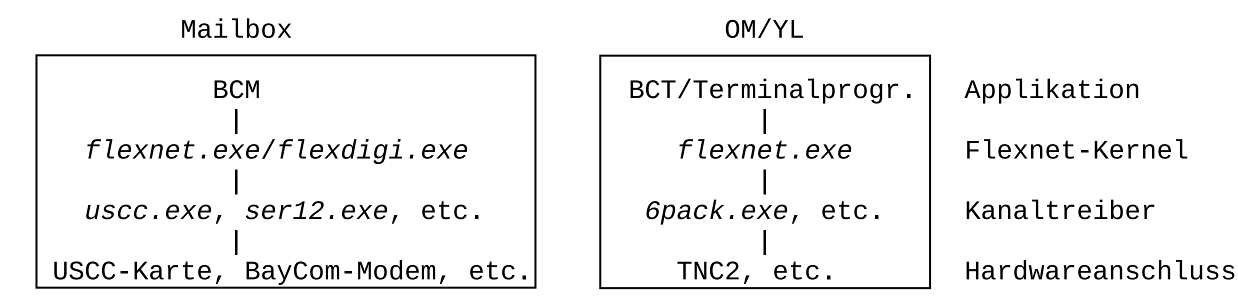
Unter Windows ist ebenfalls der Betrieb mit dem integrierten Layer2 möglich.

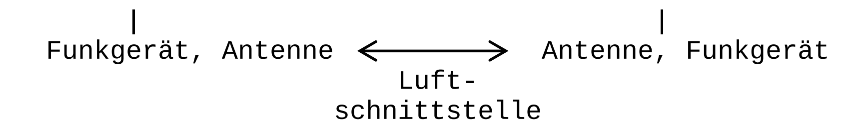
Außer den genannten, gibt es noch eine Reihe weiterer Konfigurationsmöglichkeiten. Wer glaubt eine interessante Lösung geschaffen zu haben, kann dies in einer Mail an BAYBOX @ BAYCOM schreiben.

6.1. Funkanschluss unter DOS mit PC/Flexnet

Unter DOS haben alle Konfigurationen eines gemeinsam: Die Mailbox wird auf PC/Flexnet "aufgesetzt". Je nachdem, ob PC/Flexnet gleichzeitig als Digi dienen soll oder nicht, muss *flexdigi.exe* gestartet werden. Die Kommunikation nach außen wird immer von PC/Flexnet gesteuert, jede unter PC/Flexnet einsetzbare Funk-Hardware kann also auch bei der DOS-BayCom-Mailbox verwendet werden.

Grafisch lässt sich diese Kommunikation z.B. folgendermaßen darstellen:





Die Daten werden von der OpenBCM-Mailbox an den geladenen Flexnet-Kernel übergeben. Dieser gibt die Daten an die Kanaltreiber weiter.

Im Kanaltreiber werden diese Daten je nach Hardware aufbereitet und an der entsprechenden Schnittstelle ausgegeben (z.B. serielle Schnittstelle). Im Modem werden diese Daten in über das Funkgerät übertragbare Signale umgewandelt, die Antenne strahlt die Packet-Radio Daten ab. Am anderen Ende der Verbindung befindet sich das Funkgerät des Users, die Daten gehen den umgekehrten Weg. Das BayCom-Terminal (oder irgendein anderes Terminal) stellt diese Daten am Bildschirm dar, der OM oder die YL freut sich über die neuen Nachrichten in der Mailbox.

Im Rechner muss neben dem Betriebssystem (DOS) folgendes laufen:

- Der oder die Kanaltreiber
- Der Flexnet-Kernel (evtl. mit Digi-Erweiterung)
- Die OpenBCM-Mailbox

Die Hardware bestimmt, welche Kanaltreiber zu laden sind. Es wird folgende Hardware unterstützt:

- BayCom-1k2-Modem (*ser12.exe*)
- BayCom-Parallel-9k6-Modem (*par96.exe*)
- BayCom-(U)SCC-Karte (*uscc.exe*)
- VANESSA-Karte (*vanessa.exe*)
- RMNC-CRC-KISS-Rechnerkopplung (*kiss.exe*)
- TNC2 (*6pack.exe*)
- IPX (*ipxpd.exe* bzw. *ipxn.exe*)
- AXIP (*ippd.exe*)
- Ethernet-Multicast nach BPQ (*ether.exe*)
- div. Soundkarten (*psadrvcr.exe*, *dsk.exe*, *dsk50.exe*, *dg1scr.exe*, *wss.ewe*, *wss_9k6.exe*, *sblast.exe*)

PC/Flexnet ist ein unter vielen Entwicklern verteiltes Projekt, in den Mailboxen sind in der Rubrik FLEXNET (manchmal auch RMNC) immer wieder neue Treiber zu finden.

Nachdem der Rechner gebootet wurde, ist zuerst der Flexnet-Kernel *flexnet.exe* zu laden. Anschließend wird, je nach dem, ob der Rechner gleichzeitig auch als Netzknoten dienen soll, auch *flexdigi.exe* geladen. Jetzt folgen die Kanaltreiber. Die Reihenfolge bestimmt die Port-Nummer (0..14), ist jedoch ansonsten unwesentlich. Sind alle Treiber geladen, wird Flexnet mit *flex.exe* aktiviert. Ab diesem Zeitpunkt ist Flexnet aktiv. Evtl. können oder müssen mit *fset.exe* verschiedene Parameter gesetzt werden. Zum Abschluss wird die OpenBCM-Mailbox *bcm.exe* gestartet. Mit dem Befehl VERSION kann jetzt überprüft werden, ob Flexnet erkannt wurde. Wird das Programm *flexdigi.exe* geladen, so wird im Versionsbefehl "FlexDigi" angezeigt, ansonsten "FlexNet" für Flexnet ohne Node.

Zu beachten ist, dass *flexdigi.exe* nicht frei verteilt wird, aber unter Angabe des Netzknoten-Rufzeichens bei Gunter, DK7WJ@DB0ZDF.#RPL.DEU.EU angefordert werden kann.

Womit Leute immer wieder Probleme haben: Mit *flexdigi.exe* wird die interne Struktur komplett umgebaut. Normalerweise hängen Applikationen wie *bcm* via Flexnet-Kernel direkt an den L1-Kanälen. Mit *flexdigi.exe* hängen sie an einem (virtuellen) Kanal 15 und kommunizieren nur über diesen mit der Außenwelt.

Dadurch sieht von der Parametrierung her gesehen FLEXDIGI mit BCM genauso aus wie z.B. RMNC/Flexnet (oder PC/Flexnet) mit BCM auf getrennten Rechnern. Man braucht im FLEXDIGI also auch einen Link-Eintrag zur Box auf Kanal 15. QSOs laufen immer über den Digi, was den Durchsatz etwas bremst und mehr Speicherplatz benötigt.

Daher die Empfehlung, FLEXDIGI wegzulassen, wenn keine Netzknotenfunktion gebraucht wird. Auch HOP2HOP-digipeating wird seit Version 3.3e vom Flexnet-Kernel unterstützt (siehe *flexnet.doc* und "FSET /?").

Für Sysops, die FLEXDIGI einsetzen, ist die gedruckte Sysop-Dokumentation für RMNC/Flexnet und PC/Flexnet erhältlich bei:

FlexNet-Gruppe Darmstadt
Gunter Jost, DK7WJ
Lichtenbergstr. 77
D-64289 Darmstadt
Deutschland

Achtung: Flexnet ist im CB-Funk nicht erlaubt! Daraus folgt, dass die DOS-Version der OpenBCM im CB-Funk nicht eingesetzt werden kann!

6.1.1. Parametrierung

Flexnet zeichnet sich durch nur wenige notwendige Einstellungen aus. Für jeden Kanal sind jeweils nur der MODE und das TXDELAY einzustellen, bei manchen Kanaltreibern ist auch dies nicht notwendig.

6.1.1.1. Parametrierung mit Packet-Knoten

Wird *flexdigi.exe* geladen, so werden die notwendigen Einstellungen der Datei *flexnet.fpr* entnommen, sie brauchen nicht bei jedem Start neu eingestellt zu werden. In dieser Datei werden neben dem TXDELAY und MODE für jeden Kanal auch das "ssid" sowie die eingetragenen Links gespeichert. Beim ersten Start von *flexdigi.exe* wird eine Defaultkonfigurationsdatei *flexnet.fpr* erzeugt. Diese kann entweder mit *fset.exe* oder mit dem Miniterminal *tnc.exe* den Erfordernissen angepasst werden.

Der erste Start von Flexnet könnte etwa so aussehen:

```
Eingabe                            Bemerkung
-------------------------------------------------------------------------
LOADHIGH FLEXNET.EXE 150           Flexnet-Kernel, 150 kB Puffer
LOADHIGH FLEXDIGI.EXE              Flexnet-Digi
LOADHIGH USCC.EXE /P=0x300 /I=5    Kanaltreiber für USCC-Karte
LOADHIGH SER12.EXE 2               BayCom-Modem an der COM2
FLEX.EXE                           Aktivierung von FlexNet
TNC.EXE                            Starten des Mini-Terminals
ESC C                              Connect zum Digi "FLXNET-0"
MYCALL DB0AAB 0 15                 Einstellen des Digi-Calls, SSIDs 0-15
MODE 0 9600trzums                  Einstellen der Kanal-Modes für den
MODE 1 1200cums                    Ports 1-4 der USCC-Karte (entspricht
MODE 2 19200dtrz                    den Flexnet-Kanälen 0-3)
MODE 3 19200trz
MODE 4 1200c                       ...und für das BayCom-Modem
PAR SSID 0 0                       Einstellen der Kanäle mit SSID
PAR SSID 1 1                       (= keine Exklusivlinks)
PAR TXD 10 0                       Einstellen der TX-Delays der
PAR TXD 25 1                       verschiedenen Kanäle
PAR TXD  5 2
PAR TXD  5 3
PAR TXD 25 4
MAIL DB0AAB-8                      Einstellen der lokalen Mailbox
Q                                  Verbindung zum Knoten beenden
ESC Q                              Beenden des Terminals
Y                                  Ende bestätigen
BCM.EXE                            OpenBCM-Mailbox starten
```

```
(...)                            OpenBCM-Mailbox bedienen
FLEX.EXE /U                      Flexnet aus dem Speicher entfernen
```

Vor- und Nachteile dieser Lösung sind:

- (+) Wenig Stromverbrauch, billig (Leistung pro Rechner etwa 50 W, das sind 438 kWh (etwa 60 Euro) pro Jahr.
- (-) Es steht weniger RAM für die Box zur Verfügung.
- (-) Stürzt der Digi ab, so stürzt auch die Box mit ab und umgekehrt.
- (-) Insgesamt schlechtere Stabilität sowohl von Box als auch von Digi. Die Zielsetzung beider Systeme ist unterschiedlich (Digi: Schnelle Interruptreaktionszeit, Box: Viele Dateizugriffe), so dass hier mit Konflikten zu rechnen ist.

Beispiel einer *sflex.bat* (allgemein):

```
LOADHIGH FLEXNET.EXE 150
LOADHIGH FLEXDIGI.EXE
LOADHIGH XXX.EXE (entsprechender L1-Treiber, z. B. SER12.EXE oder USCC.EXE)
FLEX.EXE
```

6.1.1.1.1. Box und Digi in einem Rechner mit USCC-Karten

- (+) Günstige Lösung, mittlere Übertragungsraten möglich, max. 38,4 kBit/s je Kanal.

Beispiel *sflex.bat* (mit USCC-Karten):

```
LOADHIGH FLEXNET.EXE 150
LOADHIGH FLEXDIGI.EXE
LOADHIGH USCC.EXE /P=0x300 /I=7
FLEX.EXE
```

6.1.1.1.2. Box und Digi in einem Rechner mit Baycom-Modem

- (+) Geringster Hardware-Aufwand, gute Lösung für erste Tests.
- (-) Nur 1200 Bit/s Übertragungsgeschwindigkeit, nur 1 Kanal.

Beispiel *sflex.bat* (mit BayCom-Modem):

```
LOADHIGH FLEXNET.EXE 150
LOADHIGH FLEXDIGI.EXE
LOADHIGH SER12.EXE 2
FLEX.EXE
```

6.1.1.2. Parametrierung ohne Packet-Knoten

Da es in diesem Fall keine Parameter-Datei gibt, müssen die Parameter MODE und TXDELAY nach jeder Aktivierung mit dem Programm *fset.exe* eingestellt werden.

Der erste Start sieht ähnlich aus:

```
Eingabe                          Bemerkung
---------------------------------------------------------------------------
LOADHIGH FLEXNET.EXE 80          Flexnet-Kernel, 80 kB Puffer
LOADHIGH KISS.EXE 2              Kanaltreiber für KISS auf COM2
                                 (z.B. Kopplung mit RMNC-Knoten)
FLEX.EXE                         Aktivierung von Flexnet
FSET.EXE MODE 0 19200c           19k2 mit RMNC-CRC
BCM.EXE                          BayCom-Mailbox starten
(...)                            BayCom-Mailbox bedienen
FLEX.EXE /U                      Flexnet aus dem Speicher entfernen
```

Sowohl das Laden, als auch das Parametrieren von Flexnet geschieht in der Startdatei *sflex.bat*. Diese Datei wird aus der *autoexec.bat* mit "CALL SFLEX.BAT" aufgerufen.

Befindet sich die Flexnet-Software im Verzeichnis *c:\flex*, so kann die Startdatei *sflex.bat* etwa folgendermaßen aussehen:

```
C:
CD \FLEX
LOADHIGH FLEXDIGI.EXE 150
LOADHIGH FLEXNET.EXE
LOADHIGH <Kanaltreiber>.exe <Parameter für Kanaltreiber>
FLEX.EXE
FSET.EXE TXD 0 10
FSET.EXE MODE 0 1200c
```

Eventuell können neben der Applikation *bcm.exe* auch andere Applikationen geladen werden. Unter DOS müssen diese Applikationen TSR's (Hintergrund-Programme) sein, unter einer anderen Oberfläche (Windows, OS/2, etc.) ggf. auch Vordergrundprogramme.

6.1.1.2.1. Anbindung der Mailbox an das Knoten-System via Ethernet

- (+) Sehr schnelle und fehlersichere Übertragung
- (+) Es können auch mehr als 2 Rechner miteinander verbunden werden (z.B. auch eine Sprachmailbox und/oder ein TCP/IP-System)
- (-) Kosten von 2 Ethernetkarten (ca. 20 € pro Karte + Abschlusswiderstände)

Beispiel *sflex.bat* (mit Novell-Netz):

```
CALL NOVELL.BAT (Lädt die Novell-Treiber)
LOADHIGH FLEXNET.EXE 50
LOADHIGH IPXN.EXE
FLEX.EXE
```

...oder eine Lösung mit Public-Domain-Treibern, hier eine Beispiel *sflex.bat* (mit PD-Packet-Treiber):

```
CALL ETHERNET.BAT (Lädt den Ethernet-Packet-Treiber)
LOADHIGH FLEXNET.EXE 50
LOADHIGH IPPD.EXE -I:0x60
FLEX.EXE
```

6.1.1.2.2. Anbindung der Mailbox an das Knoten-System über die serielle Schnittstelle

- (+) Billig, da Schnittstelle bei Mainboards oft vorhanden
- (+) Kopplung auch mit anderen Systemen (z.B. RMNC) möglich
- (+) Einfache Konfiguration
- (-) Gefahr von Übertragungsfehlern wenn kein CRC eingeschaltet ist (deshalb CRC-Kiss an beiden Enden der Verbindung und möglichst 16550 UARTs verwenden)
- (-) Es können jeweils nur zwei Rechner miteinander verbunden werden, ein Bus-System wie bei Ethernet gibt es nicht

Beispiel *sflex.bat* (KISS):

```
LOADHIGH FLEXNET.EXE 50
LOADHIGH KISS.EXE /P=0x3F8 /I=4
FLEX.EXE
FSET.EXE MODE 0 19200c
```

6.1.1.2.3. Anbindung der Mailbox an das Knoten-System mit USCC-Karten

Hier erfolgt die Verbindung über zwei USCC-Karten mit TTL-Pegeln. In früheren Konfigurationen war diese Lösung empfehlenswert. Da inzwischen aber alle

gängigen Systeme mit CRC-gesichertem KISS-Mode ausgestattet sind, ist eine Drahtkopplung über USCC eher nicht empfehlenswert, denn:

- Der RMNC ist im KISS-Mode schneller als mit HDLC (bis 115 Bit/s bei 12 MHz).
- Serielle Schnittstellen sind sehr billig und mit FIFO (16550A) erhältlich.
- RS232 hat vernünftige Schnittstellenpegel für Leitungen bis ca. 20 m, TTL ist nicht vernünftig zu übertragen.
- USCC-Karten sollten also anstatt für Drahtlinks ausschließlich für Funklinks eingesetzt werden, dafür sind sie konzipiert.

6.2. Funkanschluss unter Linux

Wie bei PC/Flexnet gibt es auch unter Linux kaum Funk-Hardware, die nicht unterstützt wird. Die meisten für DOS genannten Hardware-Kombinationen sind deshalb auch unter Linux denkbar.

Unter Linux existiert außerdem eine Menge an Packet Radio-Software, dazu gehören Knotensystem, Mailboxen, DX-Cluster, TCP/IP-Server und Convers-Systeme.

Eine Verbindung dieser Systeme untereinander ist über Pipe-Devices mit KISS oder über Loopback-AXIP realisierbar. Kernel AX.25 wird unterstützt. Es muss aber gesagt werden, dass auch unter Linux die Ressourcen nicht unerschöpflich sind. Laufen bei einem PR-Knoten viele Dienste parallel, sollten diese ggf. auf mehrere Rechner verteilt werden.

Unter Linux existiert außerdem die ausgereifte Knoten-Software XNET, außerhalb des Linux-PC kann aber auch PC/Flexnet oder RMNC/Flexnet verwendet werden.

6.2.1. Funkanschluss mit dem integrierten Layer2

Für Packetbetrieb ist in der OpenBCM-Mailbox für Linux die Funktion des Layer2 enthalten. Dieser wird über die Datei *init.l2* konfiguriert. Nur wenige Parameter müssen eingestellt werden. Eine Beispieldatei befindet sich im Kapitel "23.5. Initialisierungsdatei *init.l2*".

Der AX.25-Teil der Box kann zwei verschiedene Interfaces ansteuern:

- Zum einen funktioniert KISS-Mode an einer seriellen Schnittstelle oder an einem Pseudo-TTY zur internen Verbindung mit einer anderen Software. Wird eine serielle Schnittstelle angesteuert, so kann über den MODE- Befehl die Baudrate eingestellt werden. Diese ist jedoch nur zur Startzeit einstellbar (also vorher *init.l2* editieren) und im laufenden Betrieb nicht mehr änderbar.
  WICHTIG: Der KISS-Port ist nur für Kopplungen zu einem anderen Rechner bzw. zu einer anderen Software brauchbar. Ein KISS-TNC lässt sich damit nicht vernünftig ansteuern!
  Bei der Verbindung mit einem RMNC oder PC/Flexnet-Komponenten sollte unbedingt eine Flexnet-kompatible CRC-Prüfsumme verwendet werden. Diese wird durch ein "c" im mode-Befehl eingeschaltet. Auch erfolgt die Einschaltung automatisch, sobald ein Paket mit Prüfsumme erkannt wird. Die Parametrierung des KISS-Interfaces erfolgt in *init.l2* mit:

  ```
  assign kiss       ; Wir wollen eine KISS-Verbindung
  device /dev/ttyS0 ; über die erste serielle Schnittstelle
  mode 19200c       ; mit 19200 Baud und Flexnet-CRC
  ```

- Zum anderen wird das AXIP-Protokoll mit UDP (nicht mit Raw-IP) unterstützt. Mit dessen Hilfe ist es möglich, zu anderen Rechnern über

TCP/IP im AX.25-Protokoll zu kommunizieren. Dazu ist folgendes in *init.l2* einzutragen:

```
assign axip       ; Wir wollen eine AXIP-Verbindung
device 44.130.1.1 ; zum Rechner mit der Adresse 44.130.1.1
port 4866         ; auf dem Port 4866
```

Zusätzlich kann bei Bedarf ein Parameter "txport <portnummer>" angegeben werden. Zu diesem hin erfolgen die Aussendungen, empfangen wird stets auf dem mit "port" angegebenem Port. Dies ist jedoch ein Spezialfall und normalerweise nicht üblich.

Gedacht ist diese Anschlussmöglichkeit in erster Linie zur Verbindung mit einem PC/Flexnet-Knotenrechner. Bei diesem muss *ippd.exe* von HB9JNX benutzt werden. Es wird dabei die Option UDP verwendet, also der Parameter "/u<port>". Die IP-Adresse der Gegenstelle wird in *init.l2* unter "device" angegeben. Dabei wird ein Name angegeben, der in */etc/hosts* einer IP-Nummer zugeordnet ist, oder direkt die IP-Nummer der Gegenstation. Es sollte ein Port über 1024 benutzt werden, da hierfür auf Mailboxseite keine root-Rechte erforderlich sind. Vorgeschlagen wird auch hier der Port 4866 wie bei TCP. *ippd.exe* wird dann so aufgerufen:

```
IPPD -i:<packetintno> [-c:<numchannels>] -m:<myipaddr>
     [-u:<port>] -p:<peeripaddr> [-g:<gwipaddr>[,<gwhwaddr>]] [-r:<proto>]
```

also z.B.:

```
IPPD -I:0x60 -M:44.130.1.1 -P:44.130.1.2 -U:4866
```

Die hier verwendeten IP-Adressen sind frei erfunden und sollten koordiniert sein, falls der Rechner in irgendeinem Netz eingebunden ist. Es ist darüber hinaus sinnvoll, die Hardwareadressen (gwhwaddr=Ethernetadresse) statisch einzutragen. In einem privaten Netz sollten Internetadressen laut RFC1918 verwendet werden: 10/8, 172.16/12, 192.168/16;

Es können mehrere Instanzen von AXIP oder KISS konfiguriert werden (insgesamt max. 20). Die Kanalzuordnung erfolgt in der Reihenfolge, in der die assign-Befehle in *init.l2* auftauchen. Wird nur eine Schnittstelle (also KISS oder AXIP) verwendet, so sollte diese sinnvollerweise als erster Eintrag in *init.l2* stehen. Wird KISS ohne CRC-Prüfung betrieben, so erfolgt ein #S-Eintrag ins Syslog. KISS sollte nur mit CRC betrieben werden, deshalb immer "c" beim Modebefehl in *init.l2* eintragen.

6.2.2. Funkanschluss unter Linux über Kernel-AX.25

Die Konfiguration des Linux-Kernels wird hier nicht beschrieben. Bevor man sich daran versucht, die OpenBCM-Mailbox über das Kernel-AX.25 an die Außenwelt anzubinden, sollte man bereits mit dem Programm "call" aus den AX25-Utils erfolgreich versucht haben, einen Connect aufzubauen. Es sei hier auf das AX25- und das HAM-HOWTO verwiesen. Beides liegt normalerweise jeder Linux-Distribution bei.

Die ax25-utils sind nicht unbedingt notwendig, um die OpenBCM-Mailbox mit Kernel-AX.25 zu betreiben, wohl aber, um manche AX.25-Interfaces zu aktivieren (z.B. kissattach für KISS-Interfaces oder bpqparms für BPQ-Interfaces), außerdem empfiehlt sich das Programm *call* zum Testen der Konfiguration.

Hat man nun ein funktionierendes AX.25-Interface, gilt es zunächst herauszufinden, wie die Hardware-Adresse davon lautet. Angenommen das Interface heißt scc0, so lässt sich das mit dem Befehl "ifconfig scc0" bewerkstelligen. Die Ausgabe sieht z.B. so aus:

```
scc0      Link encap:AMPR AX.25  HWaddr DH3MB-10
          inet addr:44.130.186.1  Bcast:44.255.255.255  Mask:255.0.0.0
          UP RUNNING  MTU:256  Metric:1
          RX packets:10625 errors:1486940 dropped:0 overruns:0
          TX packets:2550 errors:0 dropped:0 overruns:0
```

Die Hardware-Adresse ("HWaddr") lautet hier DH3MB-10. Damit die OpenBCM-Mailbox nun über dieses Interface connected werden kann, ist nur ein einziger Befehl einzugeben:

ax25k_if dh3mb-10

Danach ist ein Neustart der Mailbox notwendig, um das Interface zum Kernel-AX.25 zu aktivieren. Zu beachten ist dabei, dass das Rufzeichen "dh3mb-10" nichts mit den Rufzeichen der Mailbox zu tun hat. Es ist lediglich eine Art Bezeichnung für das AX.25-Interface und wird von der Mailbox nicht weiter verwendet. Nun ist die Mailbox bereits über dieses Interface erreichbar. Um auch bei ausgehenden Connects (beim Sysop-Befehl CONNECT und bei S&F-Connects) das Kernel-AX.25-Interface zu verwenden, gibt es zwei Möglichkeiten:

- Man fügt zu obigem Befehl den Parameter "d" (für Default) hinzu, also z.B. "ax25k_if dh3mb-10 d", es wird dann bei allen ausgehenden Connects das Kernel-AX.25-Interface verwendet, d.h. ein Connect über den internen L2 ist nicht mehr möglich.
- Man gibt als ersten Parameter beim CONNECT-Befehl bzw. als drittes Feld in *fwd.bcm* einen String in der Form "ax:Hardware-Adresse" an, wobei die Hardware-Adresse des zu verwendenden Interfaces anzugeben ist. Der Befehl "CONNECT ax:dh3mb-10 DB0AAB DB0IRS" baut auf dem Kernel-AX.25-Interface mit der Hardware-Adresse DH3MB-10 einen Connect zu DB0AAB via DB0IRS auf. Innerhalb der Datei *fwd.bcm* baut die Mailbox, bei einem ausgehenden S&F-Connect zu DB0IRS, zuerst auf dem Kernel-AX.25-Interface mit der Hardware-Adresse DB0AAB-10 eine Verbindung zu DB0AAB auf, und connected dann weiter zur Mailbox DB0IRS-8.
  Wird dieser Parameter bzw. dieses Feld weggelassen, so erfolgt der Connect über den internen L2. Auch hier hat das Rufzeichen hinter "ax:" nichts mit dem eigenen Mailbox-Rufzeichen oder dem des Partners zu tun.

Abgeschaltet wird das AX.25-Interface mit "ax25k_if off". Eine Abschaltung des internen Layer2 ist derzeit noch nicht möglich, es wird allerdings daran gearbeitet.

6.3. Funkanschluss unter Windows

Unter Windows NT/2000/XP ist die verfügbare Packet Radio-Software eher dünn gesät. Hier sei die Knotensoftware XNET erwähnt, die sich momentan problemlos über AXUDP mit der OpenBCM rechnerintern verbinden (bei unterschiedlichen RX/TX-Portnummern) lässt.

Des Weiteren ist es möglich mit Flex32 und AXIP-UDP mit der OpenBCM-Mailbox zu kommunizieren. Hier kann man dann auch z.B. ein Terminalprogramm wie Paxon (http://www.paxon.de) oder Winstop (http://www.winstop.de), die ebenfalls Flex32 unterstützen, direkt auf dem Mailbox-PC installieren. Auch eine Kombination OpenBCM, XNET, Flex32 und Terminalprogramm ist denkbar.

Die Fernbedienbarkeit des Betriebssystems Windows ist jedoch recht eingeschränkt.

Wird ein rechnerexternes Netzknoten-System verwendet, so erfolgt die Verbindung über serielle Schnittstelle mit KISS-Protokoll. Für bessere Performance ist jedoch eine Verbindung über Ethernet zum Netzknoten-System zu bevorzugen, sofern das Netzknotensystem dies unterstützt.

6.3.1. Konfiguration von OpenBCM unter Windows

Bei der Windows-Version erfolgt der Packet-Radio-Anschluß wie auch bei der Linux-Version über *init.l2*.

Bei einer KISS-Verbindung über die serielle Schnittstelle wird hierbei allerdings nicht etwa /dev/ttyS0, sondern die unter DOS/Windows üblichen Bezeichnungen für die seriellen Schnittstellen angegeben, also COM1, COM2, etc. Das sieht dann in etwa so aus:

```
assign kiss       ; Wir wollen eine KISS-Verbindung
device COM1       ; über die erste serielle Schnittstelle
mode 19200c       ; mit 19200 Baud und Flexnet-CRC
```

Zum anderen wird, wie unter Linux auch, das AXIP-Protokoll mit UDP (nicht mit Raw-IP) unterstützt. Mit dessen Hilfe ist es möglich, zu anderen Rechnern über TCP/IP im AX.25-Protokoll zu kommunizieren. Dazu ist folgendes in *init.l2* einzutragen:

```
assign axip      ; Wir wollen eine AXIP-Verbindung
device 44.130.1.1 ; zum Rechner mit der Adresse 44.130.1.1
port   4866      ; auf dem Port 4866
```

6.3.2. Konfiguration von OpenBCM mit Flexnet32 unter Windows

Wenn man Flexnet32 und OpenBCM auf dem gleichen Rechner betreiben will, ist folgende Konfiguration der *init.l2* sinnvoll:

```
assign axip
peer   localhost
port   4722
txport 4721
```

Bei Flexnet32 muss dann im Control-Center zusätzlich ein AXIP-UDP-Port erstellt werden, der dann mit Port 4721 empfängt und auf Port 4722 sendet.

6.3.3. Konfiguration von OpenBCM mit XNET unter Windows

Will man XNET(NT) und OpenBCM auf dem gleichen Rechner betreiben, so ist in *init.l2* folgender Eintrag sinnvoll:

```
assign axip
peer   localhost
port   4724
txport 4723
```

Bei XNET muss in der *autoexec.net* ein AXIP-UDP-Port hinzugefügt werden, der dann mit Port 4723 empfängt und mit Port 4724 sendet. Dies ist beispielsweise ab Version 1.30 von XNET mit folgender Zeile machbar:

```
att ip0 axudp 0 1 l4723 d4724 127.0.0.1
```

6.4. Maximale Anzahl gleichzeitiger Verbindungen

Bei den meisten Mailboxsystemen (z.B. FBB, DieBox, W0RLI, MSYS) gibt es eine Grenze für die Anzahl der gleichzeitig in der Mailbox eingeloggter Stationen. Es sind dann z.B. nur 27 gleichzeitige Logins möglich.

Bei der OpenBCM-Mailbox wird für jedes Login dynamisch Speicher reserviert (ca. 5 kB unter DOS, ca. 20 kB unter Linux/Windows). Je nach Konfiguration (freier Speicher) ergibt sich eine praktische Grenze, die bei der DOS-Version bei etwa 80 gleichzeitigen Logins liegt. Unter 20 kB freien Speicher werden automatisch keine weiteren Logins zugelassen. Statisch wird die Anzahl der Logins bei DOS auf 100 Logins, bei Linux/Windows auf 200 Logins begrenzt.

Da damit eine eingeloggte Station anderen Stationen "nichts wegnimmt", gibt es bei der OpenBCM-Mailbox auch keine Reservierung bestimmter Kanäle als Sysop- bzw. Forwardkanäle: Es gibt eine ausreichende Anzahl von Kanälen, daher ist das nicht notwendig. Ein User-Timeout ist aus Sicht der Mailbox unnötig: Wer in der Box eingeloggt ist, kann schlafen so lange er will, sofern der Sysop ein Timeout nicht aus anderen Gründen aktiviert.

7. Benutzeroberfläche

7.1. DOS-Oberfläche

Der Monitor, der an den Mailbox-Rechner angeschlossen ist, ist die Systemkonsole. Die Mailbox stellt hier einen Bildschirm dar, der in drei Fenster unterteilt ist. Fast jedes Kommando zur Steuerung der Mailbox kann auch per Funk gegeben werden.

Die drei Fenster auf dem Bildschirm der Konsole sind:

- Monitor: Nach dem Start ist dieses Fenster leer. Mit "TNC TRACE -1" kann der Monitor nach dem Login eingeschaltet werden, dann wird hier der gesamte Packet-Betrieb mitgeschrieben. Wer diesen Befehl nicht bei jedem Neustart eingeben will, kann ihn auch in der Datei *startup.imp* eintragen.
- Console: Hier werden alle Systemmeldungen ausgegeben. Außerdem kommen alle Eingaben der Benutzer (nur Befehle) in diesem Fenster. Jede Meldung beginnt mit einem Buchstaben, der anzeigt, um welche Art Meldung es sich handelt:
  - Betriebsmeldungen (R=Report) werden nur am Bildschirm dargestellt, oder bei entsprechender Einstellung auch in die Datei *trace/syslog_r.bcm* geschrieben.
  - Fehlermeldungen (L=Log, S=Serious, F=Fatal, A=Abort) werden zusätzlich in die Datei *trace/syslog.bcm* geschrieben.

  Fehler der Kategorie S, F und A deuten entweder auf Softwarefehler hin, oder es sind schwerwiegende Fehler im Dateisystem. Auch grob fehlerhafte Sysopeingaben können zu Fehlermeldungen führen. Die Zahl der möglichen Meldungen ist sehr groß. Es wäre deshalb sehr mühsam, alle hier aufzuschreiben. Suspekte Meldungen bitte an BAYBOX @ BAYCOM schicken. Normalerweise findet sich jemand, der die Fehlermeldung deuten kann.
- Sysop: In diesem Fenster kann sich der Bediener in die Box einloggen. Beim Start erscheint ein Login-Prompt. An diesem kann entweder das eigene Rufzeichen angegeben werden (Login unter beliebigem Call) oder mit der Angabe von "." das in *init.bcm* eingestellte Sysop-Rufzeichen verwendet werden.

Links oben (über dem Monitor) wird die Task-Aktivität angezeigt. Jedes Zeichen entspricht einer Task in der Mailbox. Angezeigt werden folgende Zustände:

- I: Idle, also keinerlei Aktivität
- N: Neue Task, vor der ersten Suspendierung
- R: Task läuft gerade
- M: Bildschirmfenster wird aufbereitet
- U: Gesamtbildschirm wird neu eingerichtet

Rechts neben den Tasks wird der freie Speicher in Kilobytes angegeben.

Die drei Fenster auf der Konsole können mit folgenden Funktionstasten verwaltet werden:

- F1: Console-Fenster aktivieren
- F2: Monitor-Fenster aktivieren
- F3: Sysop-Fenster aktivieren
- F4-F10: Mailbox-Session 2..8
- CTRL-F3: Neues Box-Login Fenster (oder Befehl W2); Parameter: eigenes Rufzeichen; als zweiter Parameter kann das gewünschte MyCall der Box angegeben werden

- CTRL-F4: Editorfenster öffnen (oder Befehl ED <Dateiname>) In neueren Versionen der Mailbox ist der Editor nicht mehr enthalten (Speichermangel)
- CTRL-F5: (Zoom - ALT-Z) Fenster auf volle Bildschirmgröße setzen
- CTRL-F6: (Next - ALT-N) nächstes Fenster aktivieren
- CTRL-F7: (ALT-R) Fenstergröße ändern. Cursor: Move, SHIFT-Cursor: Resize
- ALT-F3: Fenster schließen (Task Kill, Vorsicht: kein Abspeichern) Jedes Fenster kann geschlossen, aber nicht unbedingt geöffnet werden (z.B. Console oder Monitor). Daher nicht unbekümmert anwenden.
- ALT-F5: DOS-Bildschirm zeigen (nur darstellen, Aussteigen mit OSHELL)
- ALT-C: Einstellen der Bildschirmfarben

Das Beenden der Software erfolgt mit ALT-X. Dabei werden alle eingeloggten Benutzer disconnected, alle Logins sauber getrennt und alle Dateien geschlossen. Sollte das nichts nutzen, so kann mit ALT-K ein harter Abbruch erfolgen, der allerdings nichts abschließt und somit offene Dinge hinterlassen kann. Normalerweise sollte jedoch auch dann nichts passieren, da das Dateisystem zu fast jeder Zeit konsistent ist.

Es ist sinnvoll, rechtzeitig vor dem Abstellen der Mailbox die Benutzer mit WALL <text> auf das Abstellen aufmerksam zu machen.

Mausbedienung: Ist am Rechner eine Maus angeschlossen, so können die Fenster auch mit der Maus bedient werden. Dabei gibt es folgende Möglichkeiten:
- Fenster anklicken: Fenster wird aktiviert kommt in den Vordergrund
- Fenster am oberen Balken anklicken und ziehen: Fenster verschieben
- Fenster rechts unten anklicken und ziehen: Größe verändern

7.2. Linux-Oberfläche

Unter Linux wird nur das Syslog ausgegeben. Ein direkter Login ist nicht möglich. Dazu kann Telnet oder das Programm *bct* verwendet werden. Auf dem normalen Bildschirm kommt die Ausgabe der Trace-Meldungen (wie am "Console"-Fenster) und des Monitors, wie auf dem Monitor-Fenster. Das Abschalten des Monitors erfolgt mit "monitor off" in *init.l2*. Zur Kommunikation ist ein Web-Browser (z.B. Netscape) empfehlenswert.

Sinnvoll ist es, diesen Bildschirm auf /dev/tty8 auszugeben, dann kann man mit ALT-F8 den Boxbildschirm anwählen, sofern auf diesem kein getty aktiv ist. Die Zugriffsrechte müssen allerdings richtig eingestellt sein, damit eine Ausgabe darauf erfolgen kann.

Die Ausgabe im Trace-Fenster erfolgt gefiltert, nicht zulässige Zeichen werden durch einen Punkt ersetzt. Damit wird verhindert, dass Zeichen durch VT100-Terminals falsch interpretiert werden. Die Zeichen 0x0d 0x0c 0x20-0xf7 werden angezeigt, andere Zeichen werden durch einen Punkt ersetzt.

7.3. Windows Oberfläche

Die Windows-Version hat derzeit keine nennenswerte Benutzeroberfläche. Es öffnet sich ein Fenster, auf dem die eingestellten Trace-Ausgaben erscheinen. Der Zeichensatz passt sich so gut wie möglich der Fenstergröße an. Weitere Schnörkel sind nicht vorhanden.

Wie bei der Linux-Version wird über ein externes Programm auf die Box zugegriffen. Ein Login ist über *telnet.exe* möglich, das bei Windows NT/2000/XP

mitgeliefert wird. Hier wird bei "Hostname" der eigene Rechnername und bei "Anschluss" der Port 4719 angegeben. Nach Verbindungsaufbau kommt ein Loginprompt wie in der DOS-Version.

8. Allgemeine Hinweise zur Wartung durch den Sysop

Das Dateisystem der OpenBCM-Mailbox ist so gut wie möglich auf Robustheit gegen Systemabstürze, Fehlbedienung und Löschen/Verfälschen von Dateien ausgelegt. Es gibt daher nur wenige Dinge, die auf keinen Fall zulässig sind bzw. danach wieder restauriert werden müssen:

- Datum/Uhrzeit: Da die generierten Dateinamen aus Datum/Uhrzeit erzeugt werden und unbedingt eindeutig sein müssen, ist es lebenswichtig, dass die Uhr im verwendeten PC funktioniert. Unzuverlässige oder falsche Uhrzeit kann zu einem gewaltigen Durcheinander, insbesondere in der CHECK-Liste führen. (Ein binäres Suchverfahren arbeitet nicht mehr richtig, wenn die Liste nicht sauber sortiert ist.) Wird während des Betriebs der Mailbox die Uhr zurückgestellt, so läuft die interne Uhr der OpenBCM-Mailbox so lange langsamer bis die eingestellte Uhrzeit erreicht ist. In dieser Zeit sollte daher die Mailbox nicht abgeschaltet werden. Ist das dennoch notwendig, so dürfen im zurückgestellten Zeitraum keine neuen Nachrichten eintreffen. Sollte die Zeit vollkommen falsch gelaufen sein, so kann mit REORG wieder Ordnung geschaffen werden. Wichtig ist, dass der Reorg nach der überlappenden Zeitperiode gestartet wird, also erst wenn sichergestellt ist, dass es keine Dateien mit neuerem Zeitstempel als die aktuelle Zeit gibt.

- Editieren von Systemdateien: Durch unkontrolliertes Editieren der Dateien *check.bcm* und *list.bcm* kann man die Box gewaltig durcheinander bringen. Ist etwas passiert, so sollte vorsichtshalber anschließend der reorg-Befehl angewendet werden. Dieser restauriert alle *list.bcm* und *check.bcm* Dateien wieder.

Auch die maschinenlesbaren Dateien *users4.bcm* und *bids2.bcm* (bzw. *bids3.bcm* bei der Linux-Version) sind davon betroffen. Sie sind nach dem Löschen unwiederbringlich weg und sollten daher regelmäßig gesichert werden!

9. Forwarding komplett einrichten

Die Einstellung des Forwardings erfolgt über die Datei *fwd.bcm*, in dem die notwendigen Daten für das Forwarding abgelegt werden. Mit dieser Datei wird festgelegt, mit welchen Boxen zu welcher Zeit Forward abgewickelt wird, und welche Mails zu welchen Boxen geschickt werden.

Die von den Nachbarboxen empfangenen Mails können durch die Forward-Datei *fwd.bcm* nicht beeinflusst werden!

In der Datei *fwd.bcm* wird für jede Nachbarbox festgelegt:

- Der Name der Nachbarbox (Rufzeichen ohne SSID)
- Die Zeiten, zu denen Forwardverbindungen aktiv aufgebaut werden
- Informationen zum Verbindungsaufbau zur Nachbarbox (Connect-Weg)
- Übertragungsoptionen (Maximale Größe einer Mail, etc.)
- Welche Mails zur Nachbarbox geschickt werden sollen

Während die ersteren Punkte leicht einzustellen sind, macht die Auswahl der Mails, welche zur Nachbarbox geschickt werden sollen, oft Schwierigkeiten.

Zur Auswahl wird normalerweise die Adresse (das ist das was beim SEND-Befehl direkt nach dem "@" steht) der Mail verwendet, dabei muss zwischen einer hierarchischen Adresse und einem Verteiler unterschieden werden.

Beispiel:

```
DF3VI @ DB0HOM.#SAR.DEU.EU
        ^Adresse (hierarchische Adresse der Zielmailbox)
^Empfängercall
```

oder:

```
LINUX @ DL
        ^Adresse (Verteiler)
^Empfängerrubrik
```

Daneben gibt es noch andere, weiter unten beschriebene Auswahlkriterien.

9.1. Die Theorie der Auswertung der hierarchischen Adresse

Eine hierarchische Adresse ist folgendermaßen aufgebaut:

```
DB0AAB.#BAY.DEU.EU
                ^ Kontinent EU steht für Europa
            ^ Land, DEU ist Deutschland.
        ^ Evtl. noch eine genauere Unterteilung nach Regionen
^ Rufzeichen der Mailbox
```

Bezeichner für Kontinente sind zwei- oder vierstellig. In Europa sollte "EU" verwendet werden.

- .AF, .AFRC: Afrika
- .AS, .ASIA: Asien
- .AU, .AUST: Australien
- .EU, .EURO: Europa
- .NA, .NOAM: Nordamerika
- .OC, .OCEA: Oceanien
- .SA, .SOAM: Südamerika
- .ANTR Antarctica (*)
- .AUNZ Australia/New Zealand (*)
- .CAFR Central Africa (*)
- .CARB Karibik (*)
- .CEAM Central America, Mittelamerika
- .EPAC Eastern Pacific (*)
- .INDI Indean Ocean inc. Indian subcontinent (*)
- .MDLT Mediterranean (*)
- .MDLE MiddleEast (z.B. Isräl)
- .NAFR Northern Africa (*)
- .NPAC Northern Pacific (*)
- .SAFR Southern Africa (*)
- .SEAS South-East Asia (*)
- .SPAC Southern Pacific (*)
- .WPAC Western Pacific (*)

Die mit (*) gekennzeichneten Kontinent-Bezeichner sind Erfindungen der TAPR, praktisch nicht verwendet, ohne Bedeutung und hier nur der Vollständigkeit halber angeführt.

Die Kenner der verschiedenen Staaten sind durch ISO 3166 definiert und bestehen aus 3 Buchstaben. In Europa gibt es folgende Kenner:

In der Praxis verwendet:

- .AUT Österreich

- .BEL Belgien
- .BGR Bulgarien
- .BIH Bosnien-Herzegowina
- .CHE Schweiz
- .CZE Tschechien
- .DEU Deutschland
- .DNK Dänemark
- .ESP Spanien
- .EST Estland
- .FIN Finnland
- .FRA Frankreich
- .GBR Großbritannien
- .GIB Gibraltar
- .GRC Griechenland
- .HRV Kroatien
- .HUN Ungarn
- .IRL Irland
- .ITA Italien
- .LTU Litauen
- .LUX Luxemburg
- .LVA Lettland
- .MKD Mazedonien
- .MLT Malta
- .NLD Niederlande
- .NOR Norwegen
- .POL Polen
- .PRT Portugal
- .ROM Rumänien
- .RUS Russische Föderation
- .SVK Slowakei
- .SVN Slowenien
- .SWE Schweden
- .TUR Türkei
- .UKR Ukraine
- .YUG (Rest-)Yugoslavien

Kurzfassung, zum Hineinkopieren in die Datei *fwd.bcm*:

```
.AUT .BEL .BGR .BIH .CHE .CZE .DEU
.DNK .ESP .EST .FIN .FRA .GBR .GIB
.GRC .HRV .HUN .IRL .ITA .LTU .LUX
.LVA .MKD .MLT .NLD .NOR .POL .PRT
.ROM .RUS .SVK .SVN .SWE .TUR
.UKR .YUG
```

Weitere Kenner:

- .ALB Albanien
- .AND Andorra
- .BLR Weißrussland
- .FRO Färöer Inseln
- .GRL Grönland
- .ISL Island
- .LIE Lichtenstein
- .MCO Monaco
- .MDA Moldavien
- .MSR Monserrat
- .SJM Svalbard/Jan Mayen

* .SMR San Marino
* .VAT Vatikan

Svalbard sind übrigens norwegische Besitzungen am Nordpolarmeer. ;-)

Folgende Kenner sind nicht zulässig:

* .SUN Soviet Union (gibt es nicht mehr)
* .CSK Tschechoslowakei (gibt es nicht mehr)
* .MAK und .MAC Mazedonien wurde .MKD zugewiesen
* .TCH Tschechien verwendet seit 1996 .CZE

Die Regionalkenner bestehen aus 4 Zeichen von denen das erste eine Raute (#) ist.

Regionalkenner in Österreich:

* .#OE1 Wien
* .#OE2 Salzburg
* .#OE3 Niederösterreich
* .#OE4 Burgenland
* .#OE5 Oberösterreich
* .#OE6 Steiermark
* .#OE7 Tirol
* .#OE8 Kärnten
* .#OE9 Vorarlberg

Regionalkenner in Tschechien:

* .#MOR Mähren
* .#BOH Böhmen

Regionalkenner in Deutschland:

* .#BAY BAYern
* .#BLN BerLiN
* .#BRB BRandenBurg
* .#BW Baden-Württemberg
* .#HB Hansestadt Bremen
* .#HES HESsen
* .#HH Hansestadt Hamburg
* .#MVP Mecklenburg-VorPommern
* .#NDS NieDerSachsen
* .#NRW NordRhein-Westfalen
* .#RPL Rheinland PfaLz
* .#SAA SAchsen-Anhalt
* .#SAR SAaRland
* .#SAX SACHen
* .#SLH SchLeswig-Holstein
* .#THR THüRingen

Bei Bezeichnern im Ausland ist zu beachten, dass die Adresse streng von rechts ausgewertet wird (Ausnahme: ein konkretes Boxrufzeichen ist bekannt). Das heißt konkret, dass eine Angabe ".#OE7" (Tirol) in einer deutschen Mailbox gar nichts bringt, weil diese von rechts in einer Adresse wie "OE7XKJ.#OE7.AUT.EU" gesucht wird. Das ".EU" darf entfallen, weil es in den eigenen Bereich fällt. ".AUT" ist jedoch außerhalb und muss deshalb angegeben werden. Der korrekte Eintrag für Tirol lautet also ".#OE7.AUT". Zusätzlich kann in eine andere Richtung noch ".AUT" angegeben werden. Da stets nach der längsten Übereinstimmung gesucht wird, werden dann alle Nachrichten nach Tirol anders verteilt als der Rest von Österreich.

Ein Sonderfall wäre die von einigen wenigen Sysops eingestellte Adresse mit .EURO am Schluss. Diese Fälle müssten einzeln abgehandelt werden (in eine Richtung .HUN.EURO, in die andere .ITA.EURO) weil die Erkennung des eigenen Kontinents hier nicht funktioniert. Die OpenBCM-Mailbox ändert intern ein ".EURO" am Schluss jedoch automatisch in ".EU", so dass in der Datei *fwd.bcm* dieser Sonderfall nicht berücksichtigt werden muss. Weitergegeben wird aber natürlich die Original-Adresse.

9.2. Die praktische Gestaltung der Einträge

9.2.1. User-Mails (hierarische Adresse)

In der gesamten Forward-Datei müssen Einträge existieren für:

- alle Kontinente, außer dem eigenen Kontinent (also außer z.B. .EU).
- alle Staaten im eigenen Kontinent, außer dem eigenen Staat (also alle europäischen Staaten außer z.B. .DEU).
- alle Regionen im eigenen Staat, außer der eigenen Region (also alle deutschen Regionen außer z.B. .#BAY).
- alle Mailboxen in der eigenen Region, außer der eigenen Mailbox (also z.B. alle Mailboxen in Bayern außer der eigenen).

Diese Einträge sind auf die Forward-Partner aufzuteilen. Es können gleiche Einträge auch für mehr als eine Box gemacht werden. Die Mail wird dann an die Box geschickt, die als erstes erreicht wird.

Es ist dabei nicht sinnvoll, Einträge nach dem Strickmuster "A* B* C* ..." zu machen, da bei einer solchen Konfiguration auch Mails mit sinnlosen Adressen (z.B. @TE0ST.NOWHERE.XYZ) weitergeleitet werden, und dann in Nachbarboxen Probleme bereiten.

9.2.2. Bulletins (keine hierarchischen Adressen)

Es gibt verschiedene Verteiler (das was beim SEND-Befehl nach dem @ steht) für Bulletins, z.B.:

- WWW: weltweit
- ALL: weltweit
- EU: europaweit
- DL: deutschsprachiges Gebiet (nicht nur Deutschland)
- ALLE: deutschsprachiges Gebiet (wenig verwendet)
- OE: Österreich
- OK: Tschechien
- OKOM, OMOK: Tschechien und Slowakei
- THEBOX: Verteilung von Sysop-Infos
- AMSAT: Satellitendaten
- BAYCOM: alle OpenBCM-Mailboxen (BCM-Software-Verteilung)
- $WP: Weiterleitung von White-Page Infos (Mybbs etc.)

Wenn die Weiterleitung eines Bulletins zur Nachbarbox gewünscht wird, so ist die entsprechende Adresse in *fwd.bcm* im Abschnitt für diese Mailbox einzutragen. Selbstverständlich ist es möglich, Bulletins an mehrere Mailboxen weiterzuleiten.

9.2.3. Alle Optionen im Überblick

In der Datei *fwd.bcm* können auch Zusatzparameter für den jeweiligen Abschnitt einer Partnermailbox angegeben werden. Das sind im nachfolgenden:

```
-b<bytes>  maximale Größe der geforwardeten Mail, es wird bei zu langen
           Mails eine Warnmail an den Adressaten geschickt
-d         (delay) Forward nur mehr zu den in crontab.bcm angegebenen Zeiten
-e         E/M-Files 4Zeilig (allerdings in 1 Frame) senden (für DB0SAO)
-f         Leerzeile vor Beginn des Forwarding senden (für TCPIP)
-i         es werden AutoBIN Mails zur Nachbar-Box geschickt, obwohl diese
           kein "D"-Feature im SID anbietet
-k         kein S&F von Bulletins im Autobin-Format
-l         kein S&F von persönlichen Nachrichten im Autobin-Format
-m         kein S&F von Bulletins im 7plus-Format
-n         kein S&F von persönlichen Nachrichten im 7plus-Format
-o         Einträge werden nach Nachrichtengröße sortiert (kleine zuerst)
-p         lässt nach dem Login den Prompt weg (für DB0SAO erforderlich)
-r         sendet AUSSCHLIESSLICH die angegebenen Boards zur Nachbarbox
-s         SID wird von connectender Box SOFORT gesendet
-t         Tracing aktiv (bei fwdtrace 2)
-u         Verbindung wird sofort abgebaut, normalerweise wird am Ende des
           S&F kurz auf noch eintreffende Nachrichten gewartet
```

Achtung: Die Verwendung der Option "-l" ist nur dann angebracht, wenn der S&F-Partner grundsätzlich nichts mit solchen Nachrichten anfangen kann und die Zielbox der Nachrichten eine Uralt-Box, z.B. veraltetes KW-Gateway, ist. Die Größenbeschränkung macht nur dann Sinn, wenn der S&F-Partner nur ASCII-Übertragung unterstützt. Intern wird die Größe eine Nachricht derzeit auf 10 MByte beschränkt. Jede Option muss einzeln durch ein Leerzeichen links und rechts getrennt angegeben werden.

9.2.4. Besondere Kriterien nur für Bulletins

Normalerweise werden Bulletins auf Grund ihres Verteilers, also ihrer Adresse (z.B. @DL oder @WW) geforwardet. Es ist aber auch möglich, nur bestimmte Rubriken von Bulletins zu forwarden, bzw. bestimmte Rubriken vom Forward auszuschließen.

9.2.4.1. Es sollen bestimmte Rubriken nicht geforwardet werden

Dazu ist im *fwd.bcm*-Abschnitt für die jeweilige Box die Rubrik mit einem Stern (*) davor anzugeben. Es wird dabei der Name vor Passieren der *convert.bcm*-Funktion verwendet.

Beispiel:

***MEINUNG *HUMOR *E**

9.2.4.2. Es sollen nur bestimmte Rubriken geforwardet werden

Dazu ist im *fwd.bcm*-Abschnitt für die jeweilige Box die Option "-R" anzugeben, und zusätzlich die Rubrik mit einem Stern (*) davor anzugeben. Es wird dabei der Name vor Passieren der *convert.bcm*-Funktion verwendet. Es werden natürlich nur jene Mails weitergeleitet, deren Verteiler ebenfalls eingetragen ist.

Beispiel:

-R *BAYBOX *SYSOP *F

9.2.5. Forward-Timeout

Kommen während des Forwards länger als im Parameter FWDTIMEOUT festgelegt keine Daten an, so wird die Verbindung abgebaut und es erfolgt ein Eintrag in *trace/syslog_r.bcm*. Die Größe des Timeouts richtet sich nach der Geschwindigkeit der Links, manchmal ist sogar ein FWDTIMEOUT von 30 Minuten zu wenig.

9.2.6. Besondere Optionen

Bei manchen Mailbox-Systemen müssen beim Aufbau der Forward-Verbindung bzw. während des Forwardings einige Besonderheiten beachtet werden, dies gilt vor allem für xNOS-Systeme. Alle Optionen sind im Kapitel "9.2.3. Alle Optionen im Überblick" bereits kurz vorgestellt worden, hier noch einmal ein paar Ausführungen zu einigen besonders wichtigen.

- Bei einigen xNOS-Boxen ist es notwendig, nach dem Aufbau der AX.25-Verbindung eine Leerzeile zu schicken, damit diese erkennen, dass es sich um eine einfache AX.25-Verbindung handelt. Dies kann mit der Option -F eingestellt werden.
- Die Option "-T" bewirkt genauso wie die Einstellung FWDTRACE 2, dass der Forward-Ablauf zu der angegebenen Box in einer Datei namens *trace/t_<rufzeichen>.bcm* mitgeschrieben wird. Dies kann bei Problemen mit dem Forward zu einer Mailbox recht hilfreich sein.
- Für Nachbarboxen, die ernsthafte Probleme mit der Anzahl der Forwardconnects haben, lässt sich der Verbindungsaufbau bei jedem Neueintreffen einer Mail abstellen. Bei Angabe der Option -D in *fwd.bcm* wird der Forward zu dieser Box nur noch zu den im *crontab.bcm* angegebenen Zeiten angestoßen. Es muss allerdings in aller Deutlichkeit gesagt werden, dass sich dadurch die Performance im Forwarding drastisch verschlechtert, und dass vor allem Bulletins dadurch nicht mehr den schnellsten Weg laufen, sondern einen zufälligen Weg, wie eben die Mail gerade ins Zeitraster fällt. Ich habe mich deswegen so lange geweigert, eine diesbezügliche Änderung in die Software einzubauen, weil ich sie für technisch absolut unsinnig halte. Ein Connect mehr oder weniger kostet im Netz so gut wie keine Resourcen, hingegen ein unsinnig gelaufenes Bulletin sehr wohl. Außerdem sollen Privatmails so schnell wie möglich ankommen und nicht darauf warten müssen, bis mal wieder ein Zeitschlitz frei ist.

9.3. Aufbau der Forward-Datei

Zuerst ein Beispiel:

```
;-----------------------------------------------------------------
; ANFANG
;-----------------------------------------------------------------
; Forward-Datei von OE3XSR.#OE3.AUT.EU
;      0         1         2
;      012345678901234567890123
;-----------------------------------------------------------------
DB0WGS - DB0WGS-8
; Boxen in der eigenen Region
 OE3XBS
; Regionalkenner (in .AUT)
 .#OE2 .#OE5 .#OE8 .#CAR .#OE9 .#TIR .#OE7
; Europa
 .BEL .BGR .BIH .CHE .DEU .DNK
 .ESP .EST .FIN .FRA .GBR .GIB
 .GRC .HRV .IRL .ITA .LTU .LUX
 .LVA .MKD .MLT .NLD .NOR .PRT
 .ROM .RUS .SVK .SVN .SWE .TUR
 .UKR .YUG
```

```
; Kontinente
 .AF .AFRC .AS .AU .AUST .CEAM .CARB .MDLE .NA
 .NOAM .OC .OCEA .SA .SOAM
; Verteiler (ohne Punkt davor)
 WW EU DL BAY OE
 BAYCOM AMSAT THEBOX
; White Page Infos für ERASE und MYBBS
$WP
;----------------------------------------------------------------
OE1XAB - OE1XAB OE3XBR
 -T
 .#OE1 .#OE4 .#OE6
 .HUN
 DL OE OEOST
 BAYCOM AMSAT
;----------------------------------------------------------------
OE3XZR - OE3XZR-8
 -T
 OE3XZR
;----------------------------------------------------------------
OK0NKT - OE3XNR / OK0NKT-12
 .CZE .POL
 WW EU AMSAT OK THEBOX
 $WP
;----------------------------------------------------------------
; ENDE
;----------------------------------------------------------------
```

Die erste Zeile eines Forward-Eintrags legt jeweils den Mailbox-Namen (Call des Forward-Partners ohne SSID), die Forward-Zeiten (wann welche Mails geforwardet werden) und den Weg zum Forward-Partner (wie der Connect aufgebaut werden muss) fest. Danach folgen beliebig viele Zeilen, die jeweils mindestens ein Leerzeichen eingerückt sind, und festlegen, welche Mails zum Partner geschickt werden (siehe oben).

Die Forward-Zeiten werden durch ein "-" festgelegt, dieses entspricht intern dem String "PAAAAAAAAAAAAAAAAAAAAAAA" und ist in den meisten Fällen sinnvoll. Es werden damit alle eingetragenen Nachrichten sofort weitergeleitet. Alternativ dazu können die Zeiten auch explizit durch einen 24 Zeichen langen String festgelegt werden. Jedes dieser Zeichen bestimmt, ob Mails innerhalb einer bestimmten Stunde weitergeleitet werden. Das erste Zeichen steht dabei für die Stunde von 0:00 Uhr bis 0:59 Uhr, das zweite für 1:00 Uhr bis 1:59 Uhr, usw. Das letzte Zeichen steht folglich für die Zeitspanne von 23:00 Uhr bis 23:59 Uhr. Die Zeichen selbst bedeuten:

- "A" (All): Es werden alle Mails weitergeleitet.
- "U" (User): Es werden nur User-Mails weitergeleitet.
- ".": Es wird nichts weitergeleitet.
- "P" (Poll): Es werden alle Mails weitergeleitet (wie bei A), zusätzlich wird zu den in *crontab.bcm* festgelegten Zeiten (normalerweise alle 30 Minuten) eine Verbindung zum Forwardpartner aufgebaut, auch wenn keine Mails für ihn vorliegen. Das kann sinnvoll sein, wenn die Nachbarbox den richtigen Weg zur eigenen Box nicht kennt und deshalb ihre Nachrichten nicht weiterleiten kann. Es ist sinnvoll, einen solchen Poll einmal pro Tag einzutragen.

Beim Connect-Weg reicht es gewöhnlich, "<ZielCall> <DigiCall>" anzugeben, es wird dann ein Connect zu <ZielCall> via <DigiCall> aufgebaut. Beispiel in einer *fwd.bcm* (bei einer ausgehenden Forward-Verbindung wird hier ein Connect zu DB0IGL-8 via DB0IRS aufgebaut):

```
DB0IGL - DB0IGL-8 DB0IRS
 <Eintrag1> <Eintrag2> ...
 [...]
```

Manchmal ist es nötig, dass der Forward-Connect mehrstufig erfolgen muss. Ist dies der Fall, so ist zwischen jeder Connect-Stufe ein "/" einzusetzen. Ein Blank davor und danach ist verbindlich. Innerhalb des Flexnet-Netzes ist ein mehrstufiger Verbindungsaufbau weder sinnvoll noch notwendig.

Beispiel in einer solchen *fwd.bcm*:
DB0WGS - DB0AAB / DB0HOB / DB0WGS-8 DB0WGS
<Eintrag1> <Eintrag2> ...
[...]

Erzeugt folgende Abfolge beim Aufbau der Forward-Verbindung:

- Aufbau einer Verbindung zu DB0AAB über das AX.25-Interface der Box (Interner L2, PC/Flexnet oder Linux-Kernel-AX.25)
- **C DB0HOB**
- Warten auf "connected to"
- **C DB0WGS-8 DB0WGS**
- Warten auf "connected to"
- Abwicklung des S&F

Die Zeichenkette zwischen den Schrägstrichen wird transparent hinter dem "C" (das ist der CONNECT-Befehl) durchgereicht, ein Eintrag wie
OE7XKJ - OE7XAR 3 / AX25 OE7XKJ-4
ist deshalb ebenfalls möglich. Hier wird zuerst ein Connect zu OE7XAR auf Port 3 aufgebaut, dann wird der String "C AX25 OE7XKJ-4" gesendet. Die Syntax der Connect-Pfadangabe wird auch beim User-S&F verwendet. Auch dort ist ein stufenweiser Aufbau der Verbindung möglich, wenn auch nicht empfehlenswert (höhere Linkbelastung, begrenzter Adressraum für Verbindungen in der Box).

Im Connect-Pfad ist es möglich, bei mehrstufigen Connects ein "=" als erstes Zeichen anzugeben. Damit wird der Text bis zum nächsten " / " transparent und ohne vorangestelltes "C" durchgereicht.

Beispiel:
F5ABC AAPAAAAAAAAAAAAAAAAAAAAA OE3XSR / =XCON F5ABC 3 / =BBS
Baut die Verbindung zu OE3XSR auf, anschließend wird "XCON F3ABC 3" ausgesendet. Wenn diese Verbindung hergestellt ist, wird "BBS" ausgesendet. Die Antworten "connected" etc. müssen erkannt werden. Wird die Antwort nicht erkannt, bitte ich um ein Log, damit kann ich die Antwort dann in den Quellcode aufnehmen. Leider sind die "connected to"-Meldungen der diversen Knoten-Systeme sehr unterschiedlich.

Ist ein weiteres "=" vorhanden, so werden zwei Befehle getrennt voneinander transparent und ohne vorangestelltes "C" ausgesendet und ohne das dazwischen eine Antwort abgewartet wird. Bei mehreren "=" werden entsprechend auch mehrere Befehle gesendet.

Beispiel:
DB0AC AAPAAAAAAAAAAAAAAAAAAAAA telnet:db0ab.dyndns.org:23 / =DB0MY =GEHEIM =C 8:DB0AC / =BBS
Baut zunächst eine Telnet-Verbindung zu DB0AB.DYNDNS.ORG auf Port 23 auf, dort wird ein "DB0MY" als Login-String ausgesendet, anschließend wird ohne eine Antwort abzuwarten "GEHEIM" als Passwort-String ausgesendet und danach auch noch "C 8:DB0AC" ohne es weitere Antwort abzuwarten gesendet. Wenn dann die Verbindung mit DB0AC hergestellt ist (also ein "connected" erkannt wurde), wird wiederum "BBS" ausgesendet.

Nun folgen in beliebig vielen Zeilen beliebig viele Einträge, die bestimmen, welche Mails mit einer bestimmten Adresse zur Nachbar-Box geschickt werden (dabei muss jede Zeile um mindestens ein Leerzeichen eingerückt sein). Es kann sich dabei um Boxrufzeichen (z.B. "DB0IRS"), Fragmente von solchen (z.B. "OE*"), Verteiler (z.B. "DL") und hierarchische Bereichsbezeichner handeln. Letztere müssen dabei mit führendem Punkt angegeben werden, also z.B. ".#BAY" oder ".AUT", denn eine Angabe wie z.B. "AUT" würde dabei als Verteiler, also @AUT betrachtet werden. Des Weiteren ist bei den Einträgen auf folgendes zu achten:

- Sollen die White-Page-Informationen auch weitergegeben werden, so ist "$WP" anzugeben.
- Bei mehrdeutigen Angaben wird stets die Angabe mit der längsten Übereinstimmung benutzt. Beispiel: OE7* und OE9* wird in einer Richtung geschickt, OE* kann in die andere Richtung geschickt werden und bedeutet

dann alles außer OE7* und OE9*. Es sollten in einer Forward-Datei allerdings so wenig Wildcards wie nur möglich verwendet werden, sofern es nicht möglich ist, ganz auf Wildcards zu verzichten.

9.3.1. Weitere Hinweise zu *fwd.bcm*

- Soll von einer Box nur empfangen, aber nicht gesendet werden, so genügt die Angabe des Rufzeichens am Zeilenanfang. Die Forward-Zeiten und der Connect-Weg kann dann weggelassen werden.
- Die Groß-/Kleinschreibung ist irrelevant.
- Ein Strichpunkt (";") an beliebiger Stelle kennzeichnet den Rest der Zeile als Kommentar.
- Innerhalb der Forward-Einträge für eine Mailbox dürfen Leerzeichen, Zeilenumbrüche und Kommentare beliebig gesetzt werden. Es ist lediglich darauf zu achten, dass jede Zeile um mindestens ein Leerzeichen eingerückt ist.
- Bei zu langen Mails (wegen gesetzter "-B" Option in *fwd.bcm*) wird eine Warnmail an den Adressaten geschickt.
- Bulletins, die bereits durch die eigene Box gelaufen sind, werden nicht angenommen.
- Die Datei wird beim Programmstart eingelesen und interpretiert. Wird es im Betrieb geändert, so kann durch das Sysop-Kommando NEW die Box dazu veranlasst werden, die Datei neu einzulesen. Nach dem Einlesen wird eine Meldung ausgegeben (Anzahl der Forward-Partner bzw. eine Fehlermeldung).
- Ändern der Datei über Funk:
  a) Datei mittels RTEXT *fwd.bcm* auslesen
     Datei editieren
     Nun wieder mit WTEXT *fwd.bcm* zurückschicken (Es empfiehlt sich, vorher ein Backup der Datei zu machen)
     Nicht vergessen, NEW einzugeben, sonst kriegt die Box von der Änderung nichts mit!
  b) Falls DF3VI_FWD_EDIT in die Mailboxsoftware einkompiliert wurde, kann auch dieser zum Editieren verwendet werden
  c) Externe Runutils von DF3VI unter DOS verwenden

9.3.2. Adressen ohne Weiterleitung

Das Boxcall "DUMMY" kann als Dummy-Forwardpartner angegeben werden. Sinn der Sache ist es, Boxrufzeichen und Verteiler anzugeben, die keine Weiterleitung bewirken sollen, trotzdem aber nicht in *trace/unknown.bcm* auftauchen sollen und keine Einträge in einer Forward-Trigger-Datei ergeben sollen.

Ein Eintrag (gedacht bei DB0AAB):

```
[...]
DUMMY - DUMMY
 DB0ABC DB0XYZ
[...]
```

bewirkt, dass

- alle Nachrichten für DB0ABC und DB0XYZ kommentarlos in DB0AAB liegen bleiben (in diesem Fall evtl. sinnvoll, weil beide Nachbarboxen derzeit z.B. nicht in Betrieb sind),
- "DUMMY" nicht im "STATUS FORWARD" auftaucht,
- für "DUMMY" keine Forward-Trigger-Dateien (z.B. *trace/u_dummy.bcm*) angelegt werden.

Die Zeiten ("-") und das Connectrufzeichen müssen aus formalen Gründen angegeben werden, damit die Ziele tatsächlich dorthin geroutet werden. Der Inhalt wird ignoriert.

9.4. Mehrere Mailboxen unter demselben Rufzeichen

Der Betrieb mehrer Boxen mit einem bestimmten Rufzeichen führt zu Problemen. Die Beschreibung hier soll helfen manche dieser Probleme zu meistern, allerdings kann trotzdem von einer solchen Konfiguration nur abgeraten werden.

Werden unter einem Rufzeichen mehrere Mailboxsysteme betrieben, so besteht oft der Wunsch nach einer Weiterleitung der Nachrichten zwischen den Systemen. Eine solche Konfiguration ist weder einfach, noch ist es für die User leicht herauszufinden, in welcher der Mailboxen nun ihre Usermails zu finden sind.

Werden mehrere Mailboxen "unter einem Rufzeichen" betrieben, so ist es wichtig, folgende Begriffe genau zu unterscheiden:

- Amateurfunkrufzeichen einer Amateurfunkanlage (z.B. DB0PV oder DL/OE3DZW/P)
- AX.25-Adresse einer Mailbox (z.B. DB0PV-8)
- Name einer Mailbox (z.B. DB0PV)
- Vollständige hierarchische Adresse einer Mailbox (z.B. DB0PV.#BAY.DEU.EU), der Name der Mailbox bildet den Anfang der hierarchischen Adresse.

Werden "mehrere Mailboxen unter einem Rufzeichen betrieben", so bedeutet das, dass

- sich die AX.25-Adressen der Mailboxen nur durch deren SSID unterscheiden.
- die Namen der Mailboxen gleich sind.
- die vollständigen hierarchischen Adressen der Mailboxen gleich sind(!!), d.h. nach außen erscheint nur eine Adresse. (Nebenwirkung: Die R:-Zeilen ändern sich je nach Absenderbox, d.h. es ist damit nicht mehr möglich von einer anderen Box aus festzustellen, welches System qrv ist; außerdem können mehrere R:-Zeilen hintereinander die selbe hierarchische Adresse besitzen.)
- ein Forwardpartner nur mit einer fixen (und nicht mehreren eigenen) Boxen Forward betreiben darf.

Damit eine solche Konfiguration überhaupt funktionieren kann, müssen folgende Voraussetzungen von den unter einem Rufzeichen betriebenen Boxen erfüllt werden:

- Forward zu einer Box mit dem selben Namen senden
- Forward von einer Box mit demselben Namen empfangen und Mails mit mehreren gleichen R:-Zeilen akzeptieren, insbesondere solche, bei denen die eigene hierarchische Adresse bereits in den R:-Zeilen aufscheint.
- Generierung unterschiedlicher BIDs zwischen den Boxsystemen.
- Möglichkeit, Usermails trotz MyBBS=Boxadresse, auf eine andere Mailbox weiterzuleiten.
- bei mehr als 2 Mailboxen, Forward immer nur im Kreis, keine Vernetzung.

Um letzten Punkt zu erfüllen, kann ein User beim MYBBS-Eintrag eine SSID mitangeben. Diese SSID ist kein Teil irgendeiner hierarchischen Adresse, sondern sie zeigt nur dann, dass der User seine MyBBS nicht in dieser Mailbox mit dem Rufzeichen hat, sondern in einer anderen Mailbox. In der Forward-Datei ist dann ein Eintrag z.B. "DB0PV-9" notwendig. Ist keine SSID angegeben und MyBBS = H-Adresse, so wird die Usermail nicht weitergeleitet. Die Eintragung einer SSID beim Forward kann nur gemeinsam mit der Option "-L" (für lokal) erfolgen.

Beispiel:

Bei DB0PV sollen zwei Mailboxen unter dem gleichen Rufzeichen DB0PV betrieben werden. DB0PV-8 sei eine OpenBCM-Mailbox, an die eine weitere OpenBCM-Mailbox unter dem Rufzeichen DB0PV-9 angebunden werden soll.

Der Eintrag in *fwd.bcm* bei der OpenBCM-Mailbox DB0PV-8 lautet dann mindestens:

```
DB0PV - DB0PV-9
 DB0PV-9
```

In *init.bcm* "boxaddress db0pv.#bay.deu.eu" und "mycall db0pv db0pv-8" eingetragen.

Bei DB0PV-9 sieht *fwd.bcm* hingegen wie folgt aus:

```
DB0PV - DB0PV-8
 DB0PV-8
```

In *init.bcm* wird ebenfalls "boxaddress db0pv.#bay.deu.eu" und "mycall db0pv-9" gesetzt.

Die Eingabe der MYBBS eines Benutzers in DB0PV-8, der seine Mails lieber in DB0PV-9 vorfinden möchte, lautet dann in DB0PV-8:

```
mybbs db0pv        (gibt generell DB0PV als Home-BBS im Netz bekannt)
mybbs -l db0pv-9   (stellt lokal zu DB0PV-9 um)
```

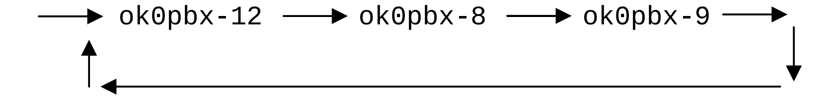
Die Eingabe der MYBBS eines anderen Benutzers in DB0PV-9, der seine Mails lieber in DB0PV-8 vorfinden möchte, lautet dann in DB0PV-9:

```
mybbs db0pv        (gibt generell DB0PV als Home-BBS im Netz bekannt)
mybbs -l db0pv-8   (stellt lokal zu DB0PV-8 um)
```

Das Vergeben von SSIDs bei Forwardeinträgen funktioniert nur bei Einträgen des eigenen Boxnamens. Es ist also keine prinzipielle Änderung der Gepflogenheiten.

Werden mehrere Boxen unter einem Rufzeichen betrieben, so ist darauf zu achten, dass die Mailboxen keine gleichen MIDs bzw. BIDs generieren. Das kann etwa dadurch sichergestellt werden, dass die Mailboxsysteme verschiedene Algorithmen zur BID-Generierung benutzen (z.B. die FBB-BBS und die OpenBCM-Mailbox). Bei der OpenBCM-Mailbox hat der Sysop keinen Einfluss auf die Generierung von BIDs/MIDs.

Da der Forwardpartner immer nur am Namen erkannt wird, ist es wichtig, dass beim Betrieb von mehr als 2 Mailboxen, diese in einem Ring in genau einer Richtung vernetzt sind, z.B. so:



Soll also eine Mail von OK0PBX-12 an OB0PBX-9 geschickt werden, so muss OK0PBX-12 diese Mail zuerst an OK0PBX-8 senden.

Wenn die Möglichkeit dazu besteht, sollte daher jede Mailbox unter einem eigenen Rufzeichen betrieben werden.

9.5. Das Einlesen der Datei *fwd.bcm*

Bei jedem Bezeichner werden, beginnend von rechts, alle Bezeichner abgeschnitten, die in der eigenen Boxadresse enthalten sind.

Beispiele bei DB0AAB.#BAY.DEU.EU:

a)

- DB0WGS.#BAY.DEU.EU
  .EU ist in der eigenen Adresse enthalten, wird also abgeschnitten
- DB0WGS.#BAY.DEU
  .DEU ist in der eigenen Adresse enthalten, wird also abgeschnitten

* DB0WGS.#BAY
  .#BAY ist in der eigenen Adresse enthalten, wird also abgeschnitten
* DB0WGS
  bleibt über, muss in *fwd.bcm* enthalten sein, da in der eigenen Region

b)

* K1XX.#BAY.NJ.USA.NA
  bleibt unverändert, da zwar .#BAY in der eigenen Adresse enthalten wäre, aber von rechts mit der Auswertung begonnen wird

c)

* .#HES.DEU
  .DEU wird abgeschnitten, weil in der eigenen Adresse enthalten
* .#HES
  überlebt

In der so eingelesenen Forward-Datei sind nur noch Bezeichner enthalten, die außerhalb der eigenen Bereiche liegen.

Außerdem wird automatisch (wenn nicht schon vorhanden) zu jeder Box das jeweilige Boxcall mit eingetragen. Das vereinfacht erstens die Auswertung, und zum anderen ist dadurch auch ein Alternativrouting zu Nachbarboxen möglich, wenn man diese auch noch woanders eintragen will.

9.6. Auswertung der Forwardadressen

Bei der Auswertung der Forwardadresse wird zunächst das erste Feld untersucht (also das Rufzeichen der Mailbox). Wenn dieses in der Forwardliste exakt gefunden wird, so wird der Rest der H-Adresse ignoriert. Da Rufzeichen weltweit eindeutig sind, ist kein sinnvoller Fall denkbar, wo dies eine falsche Weitergabe zur Folge haben könnte.

Wurde das Rufzeichen nicht gefunden worden, so werden bei der auszuwertenden Adresse von rechts beginnend alle Bezeichner abgeschnitten, die in der eigenen Boxadresse enthalten sind (Beispiel siehe oben).

Mit dieser modifizierten Adresse wird folgende Suche durchgeführt:
Jeder Bezeichner in der Forward-Datei wird mit dem Ende der modifizierten Adresse verglichen. Dabei wird in der gesamten Tabelle nach der längsten Übereinstimmung gesucht. Damit der Vergleich als erfüllt gilt, ist eine rechtsbündige, exakte Übereinstimmung erforderlich. Außerdem muss das Feld, mit dem der Vergleich durchgeführt wird, entweder mit "." beginnen, oder ein gültiges Boxrufzeichen sein (für Konstrukte wie DH5RAE.OE5XBL.AUT.EU).

Schlagen die bisherigen Vergleiche fehl, so wird nach passenden Wildcards gesucht. Diese müssen sich auf das Rufzeichen der Box beziehen (nicht auf hierarchische Adressen) und funktionieren nur bei konkreten Adressen, nicht bei Verteilern.

Kann die Adresse nicht ausgewertet werden, wird die Mail in der Liste *trace/unknown.bcm* eingetragen und bleibt in der Box liegen. Die Datei *trace/unknown.bcm* wird nicht maschinengelesen und hat daher eine für den Benutzer lesbare Form mit variabler Zeilenlänge. Vom System wird automatisch eine Mail an den Absender erzeugt. Das Utility UNK, das früher solche Mails erzeugt hat, ist damit nicht mehr notwendig.

Wird die Adresse gefunden, so erfolgen Einträge in den Forward-Listen zu allen Boxen, zu denen die Adresse geforwardet werden soll. Dazu werden, abhängig davon, ob es sich um eine User- oder um eine Rubrikmail handelt, im

Unterverzeichnis *fwd* Forward-Listen mit den Namen *u_<rufzeichen>.bcm* und *i_<rufzeichen>.bcm* geführt. Für <rufzeichen> wird das Rufzeichen der jeweiligen Mailbox eingesetzt.

In dieser Datei werden jeweils die Boardnamen und Dateinamen der zu forwardenden Mails abgelegt. Um das Board zu identifizieren reicht also der (eindeutige) Boardname, der gesamte Pfad ist nicht vonnöten. Das zweite Feld gibt die Größe der Mail an, und das optionale dritte Feld ist eine Zeitangabe im Unix-Format, welche angibt, wann diese Nachricht frühestens weitergeleitet werden soll.

Beispiel für einen Eintrag in einer Forward-Liste:
YAESU/242FE73 <Mail-Grösse> [<unix-zeit>]

Wurde die entsprechende Mail geforwardet, so wird der erste Buchstabe des Forward-Listen-Eintrags gegen ein Blank ersetzt. Sind in der Forward-Liste nur noch Einträge mit führenden Blank vorhanden, so wird sie gelöscht.

Die Einträge in dieser Datei haben variable Länge und werden jeweils mit einem Zeilenumbruch (CR/LF) abgeschlossen.

Ist in der Liste ein Eintrag für eine Mail vorhanden, die bereits gelöscht ist, so wird dieser ignoriert.

Soll eine Nachricht entgegen der bestehenden Information an eine andere Box weitergeleitet werden, so ist dies mit dem Befehl FORWARD möglich.

Vor dem Aussenden der Mails werden diese auf Plausibilität geprüft. Dabei können folgende Fehler auftreten, die dann ins Syslog eingetragen werden:

- fwds: loop von>zu box
  Nachricht ist zu oft durch die eigene Box gelaufen (Schleife) und bleibt deshalb liegen. Bulletins dürfen nur einmal angekommen sein, Usermails bis zu dreimal. Der FORWARD- oder MYBBS-Befehl übersteuert diesen Mechanismus. Der Grund liegt bei Bulletins meist darin, dass irgendwo das BID verändert worden ist und sie deshalb noch mal vorbeikommen, bei Usermails ist im Allgemeinen irgendwo das Forward gegeneinander eingestellt, so dass sich die Boxen die Nachrichten gegenseitig zuspielen. Die Meldung kann mehrfach auftauchen, da sie erst beim (bzw. anstatt) Aussenden der Mail generiert wird und daher bei jeder Box auftaucht für die die Mail bestimmt gewesen wäre.
- fwds: too old von>zu box
  Tritt bei Bulletins auf, bei denen oldestfwd überschritten wurde, die Nachricht ist zu alt und bleibt liegen.
- fwds: size von>zu box
  Die Mail ist größer als bei der "-B" Option in *fwd.bcm* angegeben und bleibt deshalb liegen.

9.7. Forwarding von White Page-Informationen

Diese Informationen werden nach dem W-Prot von OE3DZW weitergeleitet. Wird dieses Format von der Nachbarbox nicht unterstützt (kein "W" im SID), so wird das Protokoll der F6FBB-Box verwendet. Die Spezifikationen hierzu findet man im Kapitel "29. Extended White Pages Protocol (WPROT) - Spezifikation V1.0/Rev.2".

Das von der DieBox verwendete E/M-Protokoll sollte heute möglichst nicht mehr verwendet werden.

9.8. Einbuchstabige Boards

Einbuchstabige Boards sind nur mit Sysop-Rechten lesbar. Manche einbuchstabige Boards haben eine besondere Verwendung:

Board	**Software**	**Verwendung**
B	DieBox, OpenBCM	gelöschte Dateien
C	DieBox	durch KILL gelöschte Dateien
E	DieBox, OpenBCM	Remote-Erase Spezialformat (veraltet)
F	-	Sysop-Infos
M	DieBox, OpenBCM	Mybbs-Forward (veraltet)
P	Pocsag, OpenBCM	Pocsag-Nachrichten
R	-	DieBox-Softwareverteilung
T	DP-Box, DieBox, OpenBCM	TELL-Befehl
V	DP-Box	DP-Box Softwareverteilung
W	DP-Box, OpenBCM	WPROT-Informationen
X	DP-Box, WinGT	Forwardliste
Z	-	Zwischenablage für Mails

Einbuchstabige Boards sind teilweise nicht mit dem DIR-Befehl ansprechbar, weil sie durch DIR-Unterbefehle verdeckt werden ("DIR A"=DIR ALTER, "DIR B"=DIR BOARDS, "DIR N"=DIR NEWS, etc.). Diese sind dann entweder über den LIST-Befehl erreichbar oder durch Wechseln des Boards mit dem CD-Befehl und anschließendem DIR ohne Board-Angabe.

9.9. Ack-Messages

Ack-Messages werden ähnlich wie persönliche Mails weitergeleitet. Der Unterschied besteht darin, dass

- sie nur Boxen mit "A" in der SID angeboten werden.
- beim Forward keine R:-Zeilen angefügt werden.
- sie mit dem Befehl "SA" im Forward gesendet werden.
- sie intern mit dem Flag "A" gespeichert werden.

Die Eingabe des Befehls "SA" an der User-Kommandozeile ist nur möglich, wenn man als Sysop eingeloggt ist.

9.10. Anstoßen des Forwarding

Das Forwarding zu einer Box wird angestoßen, wenn eine neue Mail angekommen ist, die an die betroffene Box weitergeleitet werden muss und die Box beim letzten Versuch erreichbar war, oder jeweils zu einem im *crontab.bcm* eingestellten Zeitpunkt, wenn die Box nicht erreichbar war. Der Sysop kann mit "SF <Rufzeichen>" das Forwarding zu einer Box auch bei Bedarf anstoßen. Mit "SF ALL" wird das Forwarding zu allen Nachbarboxen angestoßen.

Es werden zuerst immer User-Mails, dann Rubrikmails und zuletzt Erase- (E @ THEBOX) und MyBBS- (M @ THEBOX) Informationen weitergeleitet. Letztere werden nicht sofort weitergeleitet, sondern nur gemeinsam mit Bulletins bzw. Usermails oder wenn das Forward durch die Datei *crontab.bcm* angestoßen wurde.

Damit das Forward jede halbe Stunde angestoßen wird, ist in der Datei *crontab.bcm* folgender Eintrag notwendig:

```
[...]
0,30      *    *   *    *      forward
[...]
```

9.11. Der Autorouter

Es besteht die Möglichkeit, eine Autorouterfunktion für das Weiterleiten von Mails in der OpenBCM-Mailbox zu aktivieren. Es muss sichergestellt sein, dass die Mailbox mit der Option _AUTOFWD in der *config.h* kompiliert wurde.

Der Autorouter versucht nur dann eine Route zu berechnen, wenn in *fwd.bcm* keine Route gefunden wurde. Man kann also den Autorouter mit dem normalen S&F in *fwd.bcm* kombinieren, indem man weiterhin bestimmte Adressen in der *fwd.bcm* einträgt. Diese haben automatisch Vorrang.

Der Autorouter wird durch den Parameter AUTOFWDTIME in der Datei *init.bcm* aktiviert. Hier gibt man die Anzahl der Tage an, die ein Boxeintrag (in *hadr4.bcm*) maximal alt sein darf, um für die Berechnungen benutzt zu werden. Ein Wert von 30 ist hier sinnvoll. Wird der Wert 0 eingegeben, ist der Autorouter deaktiviert.

Um sich einen Überblick darüber zu verschaffen, welche Routen der Autorouter eigentlich berechnet, kann man den Befehl AFWDLIST benutzen. Wird dieser gestartet, erhält man nach mehr oder weniger langer Wartezeit die Anzahl der vorhandenen Boxen und der dazu gefundenen Pfade. Genauere Informationen enthält die Datei *afwd1.tmp* im *temp*-Verzeichnis der Mailbox, welche folgendes Format besitzt:

```
    <S&F-partner> <Zielmailbox+header> <gekürzter header> <alter der info>
```

Der Autorouter berechnet den Pfad, indem er die Anzahl der dazwischen liegenden Mailboxen und der Übertragungsdauer abwägt. Als Randbedingung gilt dabei, dass mindestens 4 Mails über diesen Weg empfangen wurden.

Der Befehl AFWDLIST erstellt außerdem die Datei *afwd.bcm*, falls AUTOFWDTIME nicht auf 0 gesetzt ist, die die vom Autorouter errechneten zusätzlichen Routen enthält. Damit man als Sysop diesen Befehl nicht laufend manuell eingibt, sollte in der Datei *crontab.bcm* ein Eintrag "afwdlist" hinzugefügt werden, der dann mindestens einmal täglich ausgeführt werden sollte, also z.B.:

```
[...]
10        2    *    *     *      afwdlist
[...]
```

Das Weiterleiten von Nachrichten wird bei aktiviertem Autorouter wie folgt bewertet:

1.) *fwd.bcm*: wie üblich, nur wenn keine Adresse gefunden wurde, Punkt 2.
2.) *afwd.bcm*: Diese Datei wird vom Autorouter erstellt
3.) *hadr4.bcm*: Datenbank wird durchsucht (macht nur für ganz neue Einträge Sinn, ansonsten sind diese auch in *afwd.bcm* schon enthalten, diese Methode ist jedoch so schnell, dass man sie ruhig benutzen kann)

Mit dem Befehl AUTOPATH [-a] <Boxcall> kann man kontrollieren, ob eine Route zu <Boxcall> gefunden wurde.

9.12. Telnet-Forward unter Linux und Windows

In der Linux-/Windows-Version (nicht unter DOS) besteht die Möglichkeit einen Telnet-Forward einzurichten. Dazu wird statt dem Connectpfad die Telnetadresse des Forwardpartners in *fwd.bcm* angegeben. Der Syntax ist dem nachfolgendem Beispiel gut zu entnehmen:

```
   PWR48N AAAAAAAAAAAAAUUUUAAAAAAP telnet:pwr48n.dynu.com:23
```

Liegen Mails für die Partnerbox vor, so wird unter der Telnetadresse (diese kann auch als IP-Nummer angegeben werden) und dem definierten Port ein Forward-Connect aufgebaut. Meldet sich die Partnerbox, so wird automatisch das

Rufzeichen der eigenen Mailbox als Login ausgesendet. Benötigt die andere Mailbox ein Passwort für den Connect, so kann dieses z.B. mit

SETUSER <call> PW <passwort>

vorher definiert werden, wobei <call> das Rufzeichen der Partnermailbox ist.

Hinweis: sendet die Partnermailbox am Anfang des Forwardconnects nicht gleich ihre SID aus, so kann man in *fwd.bcm* im entsprechenden Abschnitt die Option -S angeben.

Es ist mittlerweile sowohl unter Windows als auch unter Linux jeweils ausgehender und eingehender Telnet-Forward möglich.

9.13. Datei-Forward

Wenn die Option _FILEFWD in die Mailbox einkompiliert wurde, besteht die Möglichkeit, Mails, z.B. per Email, in Dateiform mit anderen Mailboxen auszutauschen.

Zum Importieren einer solchen Datei ist der Befehl

FWDIMPORT <Boxcall> <Forwarddatei>

anzuwenden. Die Nachrichten werden so, als ob sie vom angegebenen <Boxcall> empfangen worden wären, abgespeichert.

Analog existiert für den Export einer solchen Datei der Befehl

FWDEXPORT [-<Option>] <Boxcall> <Forwarddatei>

Hier werden die für das angegebene <Boxcall> anstehenden Nachrichten in die Datei <Forwarddatei> exportiert.

Die <Option> muss dabei nicht angegeben werden, sie bewirkt folgendes:

ohne <Option>	die SID "AFHMR$" wird verwendet (FBB-Kompatibel)
-A	BCM-Erweiterungen (Autobin, WPROT-Format) werden verwendet
-D	Diebox-Erweiterungen (Autobin, E/M-Mails) werden verwendet
-S<sid>	eine eigene SID kann definiert und verwendet werden

In der Datei *fwd.bcm* ist das <Boxcall> für den Forward ganz "normal" zu definieren. Will man den Fileforward automatisieren, so ist in *fwd.bcm* anstelle eines Connectpfades

FILE[: <IMPORTFILENAME> <EXPORTFILENAME> [<EXPORTOPTION>]]

einzustellen, also z.B.:

DB0ABC AAAAAAAAAAAAAAAAAAAAAAAP FILE: DB0ABC.IMP DB0ABC.EXP D

oder

DB0ABC AAAAAAAAAAAAAAAAAAAAAAAP FILE

Für die Partnermailbox zu forwardende Mails werden dann automatisch in die Datei *bcm/fwd/export/<EXPORTFILENAME>* exportiert, und neue Mails aus der Datei *bcm/fwd/import/<IMPORTFILENAME>* importiert, wenn der Forward zu dieser Mailbox angestoßen wird (automatisch oder manuell mit SF <BOXCALL>). Die Import- bzw. Exportdatei (es werden momentan nur Dateinamen mit Großbuchstaben unterstützt) und eine mögliche Exportoption müssen nicht unbedingt angegeben werden. Werden sie weggelassen, so wird als Import-/Exportdateiname *<BOXCALL>.imp* bzw. *<BOXCALL>.exp* gewählt (die Endung ist hier jeweils in Kleinbuchstaben, damit unter Linux das Mailversendetool *sftool* ohne Patch benutzt werden kann).
Die Verzeichnisse *bcm/fwd/import* und *bcm/fwd/export* werden automatisch angelegt, wenn sie nicht existieren. Beim Befehl ST F wird eine Fileforwardpartnerbox, die in *fwd.bcm* den Eintrag FILE hat, mit dem Status "filefwd" angezeigt.

Und wie schickt man diese Forward-Dateien nun zu seiner Partnermailbox?
Unter Linux gibt es das Tool "Sftool v0.5a" (siehe http://www.wspse.de bzw. ftp://excelsior.kullen.rwth-aachen.de/pub/packet_radio/dpbox/sftool-0.5a.tar.gz) von Matthias Hensler. Ursprünglich wurde das Tool für die DPBox entwickelt, es lässt sich aber auch prima mit der OpenBCM nutzen.

Unter Windows gibt es u. a. das Freeware-Tool "Automailer v1.3.3" (siehe http://www.duodata.de/automailer), dass einem auch das manuelle Versenden der Forward-Dateien abnimmt.

10. Erkennung von 7plus Mails

Beim Empfang von Mails werden 7plus-Abschnitte automatisch erkannt. Innerhalb der Bulletin-Listen wird beim Titel dann die Zeichenfolge "(7+)" vorangestellt, beim Store&Forward wird aber selbstverständlich der ursprüngliche Titel weitergegeben.

Je nach Geschmack des Sysops können "defekte" 7plus-Teile nach dem Empfang sofort gelöscht werden, um die weitere Verbreitung zu unterbinden. Dies wird mit dem Parameter SAVEBROKEN in der Datei *init.bcm* gesteuert. Defekte Mails werden mit "(7-)" markiert.

Folgende Einstellungen sind möglich:

- SAVEBROKEN 0: Es werden nur korrekte 7plus-Mails akzeptiert, defekte Mails werden gelöscht
- SAVEBROKEN 1: Im FBB-Forward werden alle 7plus-Mails akzeptiert, im ungeschützten ASCII-Forward werden nur Nachrichten mit korrekter Zeilen-Prüfsumme und -Länge akzeptiert
- SAVEBROKEN 2: Es werden alle Mails - auch solche mit defekten 7plus-Teilen - akzeptiert

Achtung: Diese Einstellungen haben nur Einfluss auf Mails, die im Store&Forward empfangen werden. Direkt von Benutzern eingegebene Mails werden immer abgelehnt, wenn ein defekter 7plus-Teil erkannt wird.

11. Passwortfunktionen

Um eine möglichst große Kompatibilität zu anderen Systemen zu gewährleisten, wird neben dem BayCom/TheNet-Passwort-Verfahren auch das MD2/MD5-Passwort-Verfahren unterstützt. Das alte DieBox-Passwort-Verfahren wird sollte nicht mehr verwendet werden, da es bei vielen Mailboxen nach den 31.12.1999 nicht mehr funktioniert.

11.1. Das BayCom-Passwort-Verfahren

Das BayCom-Passwort-Verfahren, das dem bei TheNet ähnelt, basiert darauf, dass beide Seiten (User und Mailbox/Digi) über einen identischen String fast beliebiger Länge (üblicherweise nicht mehr als 80 Zeichen) verfügen. Bei der Passwort-Eingabe fragt die Mailbox bzw. der Digi nun nach fünf Zeichen aus diesem String. Der User sendet diese fünf Zeichen zurück. Falls die Zeichen korrekt waren, war auch die Passwort-Eingabe erfolgreich. Damit eventuelle Mitleser nicht bei jeder Passwort-Abfrage fünf Zeichen des Passworts herausfinden können, kann das Passwort in einem maximal 80 Zeichen langen Zufalls-String "versteckt" werden.

Je länger das Passwort ist, desto sicherer ist es. Die Software sollte außerdem verhindern, dass bei einer Passwortabfrage zweimal das gleiche Zeichen abgefragt wird, da sonst das Passwort relativ einfach im Zufalls-String erkannt werden kann.

11.1.1. Sysop-Identifizierung mit dem BayCom-Passwort-Verfahren

Um mit dem BayCom-Passwort Sysop-Status zu erlangen, gibt man einfach nur

PW

ein. Die Box sendet dann eine Leerzeile gefolgt von einen String wie z.B.

DB0AAB> 23 67 19 47 2

zurück. Diese Zahlen entsprechen den Stellen im Passwort-String, die der User zurücksenden muss (siehe oben). Das Sysop-BayCom-Passwort wird in der ersten Zeile der Datei *passwd.bcm* eingestellt. Der Befehl SYS bewirkt das gleiche, es wird vor der Passwort-Abfrage jedoch keine Leerzeile ausgegeben.

11.1.2. User-Login mit dem BayCom-Passwort-Verfahren

Sofern der Sysop es mit dem Befehl "USERPW 1" erlaubt hat (es handelt sich dabei um eine Einstellung in *init.bcm*), kann sich jeder User selbst ein Passwort setzen, das dann vor dem Login abgefragt wird. Das BayCom-Passwort-Verfahren wird aktiviert durch die Eingabe von:

A(LTER) LO(GINPWTYPE) BAYCOM

Die Eingabe des Passworts selbst erfolgt durch

A(LTER) PW <Passwort-String>

Das Passwort kann dabei in mehreren Schritten eingegeben werden, d.h. bei der wiederholten Eingabe dieses Befehls wird das Passwort nicht jedes Mal überschrieben, sondern der String wird jeweils an das aktuelle Passwort angehängt. Die maximale Länge des User-Passworts beträgt 39 Zeichen.

Sind nun diese beiden Einstellungen vorgenommen, so erhält man nach dem Connect der Mailbox nicht wie gewohnt den Connect-Text, sondern eine Passwortabfrage wie z.B.:

DB0AAB> 9 19 27 12 38

Wie auch beim Sysop-Passwort muss man dann mit den entsprechenden Zeichen aus dem Passwort-String antworten, die wiederum in einem Zufallsstring versteckt sein dürfen. Sind die zurückgesendeten Zeichen falsch, so wird die Verbindung getrennt, ansonsten erfolgt der Login wie gewohnt.

Durch Eingabe von "A(LTER) PW OFF" oder auch "A(LTER) PW" ohne weitere Parameter wird das Passwort gelöscht und die Passwort-Abfrage vor dem Login deaktiviert. Möchte man die Passwort-Abfrage nur vorübergehend abschalten, ohne das Passwort selbst zu löschen, so ist das durch die Eingabe von

A(LTER) LO(GINPWTYPE) OFF

möglich. Das Login-Passwort lässt sich auch soweit abschalten, dass ein erneutes Einschalten nur noch durch den Sysop möglich ist. Dies geschieht durch die Eingabe von:

A(LTER) PW DISABLE

Bei älteren Terminal-Programmen ist es oft notwendig, dass das Programm selbst eine Zeile mit einem bestimmten Befehl schickt (z.B. *pw*), damit die Passwort-Abfrage automatisch beantwortet wird. Um auch zu solchen Programmen kompatibel zu sein, ist der Befehl "ALTER PWLINE 1" gedacht. Nach dessen Eingabe wartet die Mailbox vor dem nächsten Login auf eine Zeile mit beliebigem Inhalt, bevor die Passwort-Abfrage gestartet wird. Diese Verhaltensweise lässt sich mit "ALTER PWLINE 0" wieder abstellen.

11.1.3. Store&Forward mit dem BayCom-Passwort Verfahren

Beim (User-)Store&Forward mit dem BayCom-Passwort-Verfahren wird wie beim User-Passwort der String verwendet, der mit "ALTER PW" eingestellt wurde. Ein User, der Passwort-geschützten User-Store&Forward betreiben will, wird den Passwort-

String also wie beim Login-Passwort einstellen. Beim Forward mit einer anderen (offiziellen) Mailbox jedoch wird der Sysop das mit dem Befehl

SETU(SER) <Rufzeichen der anderen Mailbox> PW <Passwort-String>

tun. Das BayCom-Passwort-Verfahren beim (User-)Store&Forward wird aktiviert mit

A(LTER) SFPWTYPE BAYCOM

beim User-Store&Forward bzw.

SETU(SER) <Rufzeichen der anderen Mailbox> SFPWTYPE BAYCOM

beim Forward mit einer anderen offiziellen Mailbox.

Alle anderen Einstellungen bzgl. des Passworts können analog zum Login-Passwort vorgenommen worden. D.h. das Löschen das Passworts erfolgt mit "ALTER PW OFF" bzw. "SETUSER <Rufzeichen der anderen Mailbox> PW OFF" und eine vorübergehende Abschaltung des Passworts ohne Löschung des Passwort-Strings selbst erfolgt mit "ALTER FWDPWTYPE OFF" bzw. "SETUSER <Rufzeichen der anderen Mailbox> FWDPWTYPE OFF".

Ist das Passwort nun auf beiden Seiten eingestellt (es muss natürlich der gleiche String und das gleiche Verfahren verwendet werden!), und connected eine Mailbox den Forward-Partner, so schickt die connectete Box nicht wie gewöhnlich nur das SID, sondern das SID gefolgt von einer Passwort-Abfrage. Das sieht dann z.B. so aus:

[OpenBCM-1.05-AB1D1FHMRW$] 7 23 49 87 34

Bei den Zahlen handelt es sich wie beim Sysop-/User-Passwort um Stellen aus dem Passwort-String. Die andere Mailbox antwortet z.B. mit

[DP-5.07-AB1D1FHMR$] j823t5cl453by9h091n9lq

also mit ihrem SID, gefolgt von einer Zeichenfolge, in der die abgefragten Zeichen des Passwort-Strings versteckt sind. Ist die Antwort korrekt, so geht es mit dem Forwarding wie gewohnt weiter, ansonsten wird eine Fehlermeldung ausgegeben, und die Verbindung getrennt.

11.2. Das MD2/MD5-Passwort-Verfahren

Beim MD2/MD5-Passwort-Verfahren wird nicht das Passwort selbst bzw. ein Teil des Passworts übertragen, sondern eine Zahl, die aus einem Zufallsstring und dem Passwort berechnet wird. Dieser Zufallsstring wird vom System, auf dem man sich identifizieren will (also z.B. der Mailbox) bestimmt. Nachdem dieser String dem anderen System mitgeteilt wurde, wird an ihn auf beiden Seiten das Passwort "angehängt". Aus der auf diese Weise entstandenen Zeichenkette wird nach dem MD2/MD5-Algorithmus (dokumentiert in RFC 1319 und RFC 1321) eine Zahl berechnet. Diese Zahl ist 128 Bit groß und wird zu dem System, auf dem die Passwort-Abfrage gestartet wurde als 32-stellige Hexadezimal-Zahl übertragen. Dort wird diese Zahl überprüft, und bei einer Übereinstimmung die Passwort-Eingabe als erfolgreich betrachtet. Das MD2-Passwort unterscheidet sich vom MD5-Passwort nur dadurch, dass der Algorithmus zur Berechnung der 128 Bit-Zahl verschieden ist, das MD5-Passwort ist dadurch minimal sicherer.

Zu beachten ist, dass der angesprochene Zufalls-String exakt 10 Zeichen lang sein muss, und nur aus Zahlen bestehen darf. Diese Einschränkung ist rein willkürlich und wurde durch die ersten Implementationen in verschiedenen Mailbox-Systemen festgelegt.

MD2 und MD5 sind übrigens ursprünglich keine Passwort-Verfahren, sondern Algorithmen zur Berechnung von CRCs, um den Inhalt zweier Dateien auf Gleichheit zu überprüfen. Das Ziel beider Verfahren ist dabei eine maximale Änderung der berechneten Zahl bei einem minimalen Unterschied innerhalb der beiden Dateien.

11.2.1. Sysop-Identifizierung mit dem MD2/MD5-Passwort-Verfahren

Um mit dem MD2/MD5-Passwort Sysop-Status zu erlangen, gibt man einfach nur

MD2 (Beim MD2-Passwortverfahren)

bzw.

MD5 (Beim MD5-Passwortverfahren)

ein. Die Box sendet dann eine Leerzeile gefolgt von einem String wie z.B.:

DB0AAB> [1234567890]

zurück. Die Zahlen innerhalb der eckigen Klammern sind der Zufallsstring, der an das Passwort angehängt wird. Daraus wird mit dem MD2- bzw. mit dem MD5-Algorithmus eine Zahl berechnet, die dann zur Mailbox geschickt wird (siehe oben). Das Sysop-MD2-Passwort wird in der zweiten Zeile in der Datei *passwd.bcm* eingestellt, das MD5-Passwort in der dritten.

11.2.2. User-Login mit dem MD2/MD5-Passwort-Verfahren

Der Passwort-String wird wie beim BayCom-Passwort-Verfahren mit "ALTER PW" eingestellt. Die Aktivierung des MD2/MD5-Passwort-Verfahrens beim User-Login erfolgt mit

A(LTER) LO(GINPWTYPE) MD2

bzw.

A(LTER) LO(GINPWTYPE) MD5

Die Passwort-Abfrage und die Berechnung der Antwort erfolgt analog zur Sysop-Identifikation.

11.2.3. Store&Forward mit dem MD2/MD5-Passwort-Verfahren

Der Passwort-String wird wie beim BayCom-Passwort-Verfahren mit "ALTER PW" eingestellt. Die Aktivierung des MD2/MD5-Passwort-Verfahrens beim (User-) Store&Forward erfolgt mit

A(LTER) FW(DPWTYPE) MD2 bzw. MD5 (beim User-Store&Forward)

bzw.

SETU(SER) <Rufzeichen der anderen Mailbox> FW(DPWTYPE) MD2 bzw. MD5

beim Forward mit einer anderen offiziellen Mailbox.

Die Passwort-Abfrage beim Store&Forward erfolgt analog zum BayCom-Verfahren, die Zufalls-Zahlenfolge wird in eckigen Klammern hinter dem SID mit einem Leerzeichen getrennt übertragen.

11.3. Änderungen nach erfolgreicher Sysop-Identifizierung

Nach der erfolgreichen Sysop-Passworteingabe stehen zusätzlich zu den User-Kommandos auch alle Sysop-Funktionen zur Verfügung. An den User-Kommandos ändert sich folgendes:

- Der Sysop ist Eigentümer von allen Mails, das bezieht sich auf die Befehle ERASE, UNERASE, SETLIFE, FORWARD und TRANSFER.
- Alle Eingaben des Sysops werden in der Datei *trace/cmdlog.bcm* gespeichert.
- Beim ERASE-Befehl kommt die Option -F hinzu. Wird diese Option angegeben, wird die Nachricht nicht nur lokal gelöscht, sondern es wird auch noch eine Remote-Erase-Mail erzeugt. Dies sollte allerdings nur bei wirklich offensichtlich AFu-schädlichen Mails angewendet werden, um den Ruf einer Zensur zu vermeiden.

Merke: Das persönliche Empfinden, dass eine Mail gelöscht werden soll, unterscheidet sich im Allgemeinen von einem Sysop zum nächsten. Dem sollte dringend Rechnung getragen werden und nur dann vom globalen Löschen Gebrauch gemacht werden, wenn keinerlei Zweifel besteht, dass die Nachricht in den Abfall soll.

- Boards mit einstelligen Namen wie F oder T können gelesen und beschrieben werden. Einstellige Boards haben teilweise eine besondere Bedeutung.
- Mit Sysop-Berechtigung sind auch gelöschte Nachrichten lesbar, ein UNERASE ist daher zum Lesen nicht erforderlich.

Bezüglich des Sysop-Status sind folgende Dinge zu beachten:

- An der DOS-Mailbox-Console ist man nach dem Login immer Sysop. Das gleiche gilt, wenn man sich bei einer Linux/Windows-OpenBCM-Mailbox von einem Rechner aus einloggt, dessen Hostname oder/und IP-Adresse in der Datei *rhosts.bcm* angegeben ist. Ferner kann man die Datei *asysop.bcm* zur Autosysop-Funktion nutzen.
- Will man den Sysop-Status wieder ablegen, so ist das durch die Eingabe des Befehls PW OFF möglich.
- Wenn die Box DISABLED ist, dann ist ein Einloggen nur durch die Eingabe des Passworts möglich.

11.4. Passwort-geschützter Store&Forward zwischen verschiedenen Mailbox-Systemen

Wird zwischen verschiedenen Mailbox-Systemen geforwardet, so ist die Verwendung des MD5-Passwort-Vefahrens sinnvoll.

11.5. Die Prinzipien der verschiedenen Passwort-Verfahren

Da die Passworteingabe von jedermann mitgelesen werden kann, erfolgt nicht, wie bei anderen Systemen (z.B. UNIX) üblich, die Eingabe einer "geheimen" Zeichenkette, sondern es werden entweder nur Teile des Passworts abgefragt (BayCom-Verfahren), oder das Passwort wird vor der Übertragung auf eine bestimmte Weise "verschlüsselt", wobei diese "Verschlüsselung" natürlich bei jeder Passwort-Übertragung verschieden sein muss (MD2/MD5-Passwort-Verfahren). Dabei wird der String, mit dem das Passwort "verschlüsselt" werden soll, von der Mailbox bestimmt. Dieser String muss absolut zufällig sein, darf sich also keinesfalls innerhalb eines kurzen Zeitraums wiederholen und die Verschlüsselung darf sich auf keinen Fall umkehren lassen (sonst kann ja jeder das Passwort berechnen).

11.5.1. Das Prinzip des MD2/MD5-Passwort-Verfahren

Wie oben schon beschrieben wird hier dem Passwort ein von der Mailbox erzeugter Zufalls-String vorangestellt und aus der entstandenen Zeichenkette eine 128 Bit-Zahl berechnet. Diese wird dann zur Mailbox übertragen, wo anschließend überprüft wird, ob auf beiden Seiten das gleiche Passwort vorliegt.

Da bei dieser Verfahrensweise nie das Passwort selbst oder ein Teil des Passworts übertragen wird, kann ein Mitleser auch nie einen Teil des Passworts herausfinden. Auch die 128 Bit-Zahl selbst, kann ein "Einbrecher" unmöglich erraten. Die einzige theoretische Möglichkeit wäre, mit einer mitgelesenen Kombination aus Zufallsstring und zur Mailbox gesendeten Zahl alle Passwort-Möglichkeiten durchzuprobieren, und so das Passwort zu "knacken". Da es aber

bereits bei einem fünf-stelligen Passwort über eine Milliarde Kombinationsmöglichkeiten gibt, kann ein MD2/MD5-Passwort durchaus als "unknackbar" angesehen werden.

11.6. Alle Passwortverfahren für User und Forward im Überblick

Alle Benutzer können sich selbst ein AX25-Passwort einstellen, sofern dies vom Sysop freigegeben wurde. Zum Login bei HTTP und POP3 via Amateurfunk ist als Passwort der eigene Vorname einzugeben. Beim Zugriff über Drahtnetze (im allgemeinen Internet) und beim CB-Funk ist ein vom Sysop einzustellendes TTY-Passwort notwendig.

Zugriff	Mögliche Passwortverfahren	Befehle
Allgemein:		
AX25	**BayCom, MD2, MD5, Priv**	**a pw, a pwline, a loginpwtype, pw**
AX25-S&F	**BayCom, MD2, MD5, Priv**	**a pw, a sfpwtype**
Sysop	**BayCom, MD2, MD5, Priv**	**sysop, priv, md2, md5**
nur DOS:		
Modem	**TTY-Passwort**	**a ttypw (nur Sysop)**
nur Linux und Windows:		
Telnet	**TTY-Passwort**	**a ttypw (nur Sysop)**
Telnet-S&F	**BayCom, MD2, MD5, Priv**	**a pw, a sfpwtype und a ttypw**
POP3-Afu	**Vorname**	**a name**
POP3-Internet	**TTY-Passwort**	**a ttypw (nur Sysop)**
SMTP-Afu	**Vorname**	**a name**
SMTP-Internet	**TTY-Passwort**	**a ttypw**
HTTP-Afu	**Vorname**	**a name**
HTTP-Internet	**TTY-Passwort**	**a ttypw (nur Sysop)**

PRIV wird verwendet, sobald eine Datei *<call>.bcm* bzw. *priv.bcm* existiert. Das Sysop-Passwort wird in der Datei *passwd.bcm* abgelegt. Falls das AX25-Passwort die Zeichenkette "DUMMY" (in Grossbuchstaben) enthält, ist ein Login via HTTP/POP3 trotzdem möglich. Dort wird dann als Passwort der Vorname abgefragt, der zuvor mit ALTER NAME über das AX25-Login gesetzt wurde. SMTP ist im Allgemeinen nur möglich, wenn zuvor POP3 ausgeführt wurde oder ALTER UNSECURESMTP 1 gesetzt ist.

12. PW & Hold/Reject Funktionen der Mailbox

Seit dem Jahr 2000 ist die OpenBCM mit einem erweitertem PW & HOLD/REJECT-System ausgestattet, doch leider wird dies bislang so gut wie gar nicht angewendet.

PW & HOLD/REJECT ist ein zweiteiliges System, mit dem sich die Sicherheit der Benutzer gegen Rufzeichenmissbrauch bedeutend erhöhen lässt.

Dies geschieht durch zweierlei Maßnahmen:
- Schutz für Benutzer, die noch kein Passwort benutzen
- Hold-Mechanismus für Nachrichten aus ungeschützter Quelle

a) Passwortverfahren

Das geänderte Passwortverfahren wird mit dem Parameter "USERPW 2" aktiviert.

Damit gelten automatisch eine Reihe Schutzmassnahmen für Benutzer ohne Passwort, was den Zugriff auf Nachrichten betrifft:

- kein unauthorisiertes Löschen
- kein Schreiben von (gefälschten) Nachrichten mit Forward
- kein Lesen von fremden Usermails
- kein Verstellen von wichtigen Parametern (wie MyBBS)

In der Datei *runutil.bcm* dient der Parameter "-P" dazu, dass Programme nur von Benutzern mit User-PW aufgerufen werden können.

In der Datei *msg\pwonly.<sprache>* kann man einen Hinweistext für Benutzer ohne Passwort hinterlegen, der beim Login gesendet wird. Hier sollte man insbesondere darauf hinweisen, wie man ein Passwort bekommt.

Im Gegensatz zum bisherigen Verhalten, wo unter Rufzeichenmissbrauch jeder jedem ein Passwort setzen (und somit aussperren) kann, wird das Passwort nun vom Sysop eingerichtet. Mit den Befehlen PWGEN und SETPW lässt sich eine automatische Passwort-Vergabe durch den Sysop einrichten:
Dazu wird mittels PWGEN eine Datei *userpw.bcm* mit 81 Passwörtern vorbereitet, die dann mit "setpw <user> <nr>" Usern zugeordnet werden können. Diese Datei *userpw.bcm* sollte man auch zu Hause haben, um das zugeteilte Passwort dem User mitteilen zu können (natürlich nicht über Funk auslesen!).

Mit dem Parameter "USERPW 3" wird das User-Passwort nicht mehr beim Login, sondern erst später abgefragt - analog zur DPBox. So wird das User-Passwort nur noch zum Schreiben und Löschen von Mails sowie zum Verstellen von Einstellungen benötigt. Vorteil: ein vergessenes Passwort sperrt einen nicht gleich aus.

b) HOLD-Mechanismus
Der HOLD/REJECT-Mechanismus wird über die Datei *reject.bcm* gesteuert. Zum bequemen Verwalten der Einstellungen gibt es den REJECTEDITOR. In *reject.bcm* kann man parametrieren, wie mit Nachrichten verfahren wird, die ohne Passwort gesendet werden oder über ungeschützte S&F-Wege hereinkommen. Mindestens ein Eintrag "P .B" (kein Bulletin-Send ohne AX25-PW) sollte vorhanden sein. Alle Optionen, die in *reject.bcm* angewendet werden können, sind im Kapitel "5.10. Reject und Hold (*reject.bcm*)" beschrieben.

Wie lange angehaltene Nachrichten nicht geforwarded werden, wird über den Parameter HOLDTIME in Stunden eingestellt (typisch: 48 Stunden = 2 Tage).

Mit DIR HOLD lassen sich angehaltene Nachrichten aufzeigen. DIR HOLD funktioniert genauso wie DIR NEWS oder CHECK abhängig vom LastCheck-Timer.

Mit "HOLD -u" bzw. "FORWARD -h" kann der Sysop auf Hold gesetzte Mails wieder in den Forward geben.

13. M(ail)-Filter Funktion

Es ist möglich mit Hilfe eines externen (Mailfilter-)Programms eingegebene Mails in der Mailbox auf bestimmte Dinge hin zu untersuchen, und dann resultierend vom Ergebnis eine Aktion mit der Mail ausführen.

Das externe Programm kann vom Sysop mit dem Befehl M_FILTER <Programm> eingestellt werden.

Das <Programm> kann dann die Nachricht untersuchen, z.B. auf unerwünschte Schlüsselwörter, oder ähnliches. Das <Programm> muss einen Rückgabewert übermitteln, von dem abhängig dann folgendes geschieht:
0: es passiert nichts
1: die Nachricht wird gelöscht
2: die Nachricht wird nicht weitergeleitet (Hold)

Unter Linux wird der Rückgabewert nicht ausgewertet, hier kann M_FILTER also nur dazu verwendet werden, ein externes Programm aufzurufen.

Der M_FILTER Aufruf wird in der Datei *trace\m_filter.bcm* protokolliert.

Anmerkung: Ist das <Programm> nicht vorhanden, oder ist der Pfad falsch eingestellt, kommt es beim Senden von Mails zu einer Ausgabe im Syslog. Ferner könnte unter Linux auch noch die Möglichkeit bestehen, dass die

Dateizugriffsberechtigungen des M_Filter-Programmes falsch gesetzt sind (es muß mindestens 'rx' gesetzt sein). Um das Mailfilterprogramm zu deaktivieren muß nur die Option M_FILTER OFF in der *init.bcm* gesetzt werden.

Hinweis: Der Funkrufserver unter Linux von Holger Flemming, DH4DAI und Jens Schoon, DH6BB (siehe im Internet "http://sourceforge.net/projects/ham-pager") verwendet ebenfalls die M_FILTER Schnittstelle. Der Pager gibt jedoch immer 0 (es passiert nichts) an die Mailbox zurück.

14. Sicherung der Mailbox-Daten

14.1. Backup unter DOS

In älteren Ausgaben der Dokumentation wurde empfohlen ein Backup mittels eines Batches zu machen. Genau dieses Backup führte aber oft zu Problemen (User werden disconnected weil Flexnet zu lange nicht gepollt wurde, Box gerät in den Watchdog-Reset weil der DOS-Aufruf zu lange dauerte etc.).

Deshalb ist es besser, gelegentlich ein Backup der gesamten Platte zu machen als täglich Dateien umzukopieren. Hier ist z.B. die kommerzielle Software "Norton Ghost" oder "Powerquest Drive Image" zu empfehlen.

Generell gilt: Die Dateien u*sers4.bcm* und *hadr4.bcm* werden zwar von der Mailbox erzeugt, können aber nicht aus anderen Daten restauriert werden, falls sie verlorengehen. Sie sollten deshalb regelmäßig gesichert werden.

14.2. Backup unter Linux

Unter Linux kann ein "save"-Aufruf für *save.imp* im *crontab.bcm* angeführt werden. Ein *save.imp* kann etwa folgendermaßen aussehen:

```
oshell mkdir backup
oshell mv -f backup\bu002.tgz backup\bu003.tgz
oshell mv -f backup\bu001.tgz backup\bu002.tgz
oshell tar zcvf backup\bu001.tgz *.bcm
```

Während der Abarbeitung des Backups läuft die Mailbox im Gegensatz zu DOS ohne Unterbrechung weiter.

Auch hier ist ein kommerzielles Partitions-Backuptool wie z.B. "Powerquest Drive Image 2002" zu empfehlen, das auch Linux-Partitionen backup-pen kann. Dann erspart man sich im Festplatten-Fehlerfall auch die Neuinstallation und vor allem Neukonfiguration des kompletten Linuxsystems incl. Mailbox.

14.3. Backup unter Windows

Unter Windows kann ebenso wie bei Linux ein "save"-Aufruf im *crontab.bcm* angeführt werden. Die Datei *save.imp* kann etwa folgendermaßen aussehen:

```
oshell mkdir backup
oshell move backup\bu002.zip backup\bu003.zip
oshell move backup\bu001.zip backup\bu002.zip
oshell pkzip -9 backup\bu001.zip *.bcm
```

Anstelle von *pkzip.exe* kann natürlich auch ein anderes Packprogramm verwendet werden. Da kein Packprogramm Bestandteil des Betriebssystems Windows ist, muss dies auf alle Fälle nachträglich installiert werden. Zum fehlerfreien Aufruf bietet es sich an, den Packer in die PATH-Variable aufzunehmen.

Während der Abarbeitung des Backups läuft die Mailbox im Gegensatz zu DOS ohne Unterbrechung weiter.

Auch unter Windows ist ein kommerzielles Partitions-Backuptool wie z.B. "Powerquest Drive Image 2002" zu empfehlen. Dann erspart man sich im Festplatten-Fehlerfall auch die Neuinstallation und vor allem Neukonfiguration des kompletten Betriebssystems incl. Mailbox.

15. Zeitgesteuerte Prozesse

Gesteuert wird die gesamte Zeitverarbeitung durch die Datei *crontab.bcm*. Diese Datei ist im Format sehr ähnlich der crontab-Datei unter UNIX/Linux.

Die folgende Beispiel-Datei zeigt den Aufbau für zeitgesteuerten Jobs:

```
# Timing-File for OpenBCM-Mailbox - CRONTAB.BCM
#
# ranges:
# minute: 0-59    hour: 0-23     day: 1-31     month: 1-12
# weekday: 0-6    (0:sun 1:mon 2:tue 3:wed 4:thu 5:fri 6:sat)
#
#minute   hour  day month weekday   command
*           *     *   *     *       minute
*/5         *    *    *     *       fiveminute
0,30        *     *   *     *       halfhour
1,31        *     *   *     *       beacon
3,33        *     *   *     *       forward
5           2     *   *    1,4      postfwd
5           3     *   *     *       purge
35          3     *   *     *       save
5           4     *   *     2       reorg
0,15,30,45  *     *   *     *       quarter
2           0     *   *     *       hour0
2           1     *   *     *       hour1
[...]
2          22     *   *     *       hour22
2          23     *   *     *       hour23
```

Kommentarzeilen beginnen mit einem Zeichen Strichpunkt (;) oder einer Raute (#) in der ersten Spalte, jedes Zeichen ungleich 0-9 oder * ist ebenfalls zur Kennzeichnung eines Kommentars zulässig.

In den linken 5 Spalten werden Zeiten angegeben. Hier kann jeweils entweder eine Zeit stehen, oder mehrere durch "," getrennt, oder Zeitbereiche in der Form "5-12" oder jede beliebige Kombination die nicht länger als 30 Zeichen ist, also z.B. "1,2,4-7,5,6-12,16". Ein "*" steht für "jede Zeit ist gültig". Mit z.B. "*/5" ist es möglich z.B. alle 5 Minuten Aufgaben ausführen zu lassen. Als Zeitpunkt werden Minuten, Stunden, Tag im Monat, Monat und Tag der Woche angegeben. Bei letzterem bedeutet 0=Sonntag und zählt bis 6=Samstag. Alle Zeitangaben müssen erfüllt sein, damit das angegebene Programm ausgeführt wird ("UND"-Verknüpfung). Sollten mehrere Zeitmodelle "ODER"-verknüpft werden, so kann dasselbe Programm in mehreren Zeilen angeführt werden.

Das ausgeführte Kommando ist entweder eine Batch-Datei (wird auf Betriebssystem-Ebene ausgeführt und muss im Dateisystem die Erweiterung *.bat* haben), oder eine Import-Datei (wird auf */bcm*-Ebene ausgeführt und muss im Dateisystem die Erweiterung *.imp* haben). Außerdem werden einige reservierte, interne Kommandos ausgeführt. Diese sind:

purge
: Startet das Ausputzen des kompletten Dateisystems (wurde bis zur Version 1.34 zu "purgehour" gestartet).

forward
: Stößt das Forwarding zu benachbarten Boxen an (wurde bis zur Version 1.34 zur vollen und halben Stunde gestartet).

beacon
Stößt das Erzeugen einer Mail-Bake an (wurde bis zur Version 1.34 in "mailbeacon"-Intervallen gestartet).
postfwd
Leitet liegen gebliebene Usermails weiter.
reorg [opt]
Startet die Dateisystem-Reorganisation. Wenn [opt] weggelassen wird, wird ein kompletter Reorg gemacht. Ansonsten kann ein Teil-Reorg angegeben werden.

Sind mehrere Bedingungen in *crontab.bcm* gleichzeitig erfüllt, so werden sie in der Reihenfolge der *crontab.bcm* abgearbeitet. Wenn unter DOS ein Batch-Programm (also ein externer DOS-Aufruf) mehr als eine Minute dauert, kann es passieren dass andere Jobs, die unter derselben Zeit, aber weiter hinten stehen, nicht mehr aufgerufen werden, weil inzwischen der Aufrufzeitpunkt überschritten wurde. Unter Linux tritt dieses Problem nicht auf, da die Ausführung der Mailbox durch einen Shell-Aufruf nicht unterbrochen wird.

Interne Kommandos (PURGE, FORWARD, etc.) sowie IMPORT-Dateien werden nebenläufig ausgeführt, d.h. blockieren sich nicht gegenseitig, sondern starten wirklich zur gleichen Zeit. Wenn solche Überschneidungen unerwünscht sind, so ist dies durch entsprechendes Timing oder SLEEP-Kommandos in den entsprechenden Import-Dateien zu verhindern.

Die Namen der BATches und IMPorts brauchen nicht "hour17" zu lauten, es ist sinnvoller der Anwendung entsprechende Namen zu verwenden, etwa *mkmstat.imp* für "make-monthly-statistics".

16. Automatische Server

16.1. Filesurf

In der OpenBCM-Mailbox ist die Funktionalität eines File-Servers (kurz Filesurf genannt) enthalten. Dieser dient in erster Linie dazu, um den Benutzern Dateien zur Verfügung zu stellen, wobei folgende Vorteile gegenüber Dateien entstehen, die in der gewöhnlichen Bulletin-Struktur gespeichert sind:

- Der Benutzer kann selbst entscheiden, mit welchem Protokoll er eine Datei ausliest (AutoBIN, YAPP, Didadit etc.)
- Die Dateien können besser sortiert werden, weil Verzeichnisse (unter Beachtung der Grenzen des zugrunde liegenden Betriebs- und Dateisystems) beliebig geschachtelt werden können
- Das Mailbox-System selbst wird nicht belastet, weil die Dateien nicht verwaltet werden müssen, da sie ohnehin keinerlei Lifetime unterliegen und auch nicht geforwardet werden müssen o. ä.

Um die Dateien zur Verfügung zu stellen, müssen sich die Dateien unterhalb eines oder mehrerer Verzeichnisse auf der Festplatte befinden. Diese Verzeichnisse werden dann durch den Parameter FSPATH in der Datei *init.bcm* für die Benutzer freigegeben. Dabei werden die verschiedenen Pfade jeweils durch ein Leerzeichen getrennt. Möchte man also z.B. die Verzeichnisse */filesurf* und */cdrom* freigeben, so gibt man in der Mailbox einfach
FSPATH /filesurf /cdrom
ein. Dabei ist zu beachten, daß grundsätzlich alle Dateien und Unterverzeichnisse unterhalb dieser Verzeichnisse freigegeben werden. Unter Linux ist allerdings zusätzlich darauf zu achten, daß der User "bcm" auf die gewünschten Dateien Zugriff hat.

Unter DOS und Windows dürfen auch Laufwerksbuchstaben angegeben werden, z. B.:
FSPATH E:\ D:\FILESURF

Der Benutzer kann dann wie an der DOS-Kommandozeile üblich mit dem Laufwerksbuchstaben gefolgt von einem Doppelpunkt (also z.B. *E:*) auf das gewünschte Laufwerk wechseln, dabei wird dann in das erste Verzeichnis gewechselt, das auf dem gewählten Laufwerk freigegeben ist.

Will man den Benutzern den Schreibzugriff auf ein Verzeichnis gewähren, so ist dem Verzeichnis ein "+" voranzustellen. Um gegenüber obigem Beispiel also z.B. noch das Verzeichnis */user* mit Schreibzugriff freizugeben, ist folgendes einzugeben:
FSPATH /filesurf /cdrom +/user

Um in den Filesurf-Modus zu gelangen, ist der Befehl FS einzugeben, verlassen wird er mit QUIT. Die Bedienung ist sehr intuitiv und erfolgt ähnlich wie bei der DOS-Kommandozeile unter der Linux-Shell. Eine Liste aller Befehle befindet sich im Kapitel "24.4. Übersicht über alle Befehle im Filesurf".

Achtung: Der Filesurf ist nur dann in der Mailbox verfügbar, wenn er als Option beim Kompilieren der Mailbox aktiviert wurde!

Der Filesurf kann auch per TCPIP unter Windows und Linux über den eingebauten FTP-Server auf Port 8021 erreicht werden (siehe hierzu auch "19.6. Zugang über FTP unter Linux und Windows").

16.2. Mailing-Listserver

In der OpenBCM-Mailbox enthalten ist auch ein Mailing-Listserver (auch kurz Mailserver genannt).

Dieser Mailserver dient zum Verteilen einer persönlichen Nachricht an mehrere vorher definierte Empfänger in Form einer Mailserverliste.

Soll eine Nachricht an eine bestehende Mailserverliste geschickt werden, so ist die Nachricht an die Adresse der Mailbox zu senden mit dem Titel:
s <Boxcall> mailto <Mailserverliste> <Titel>

Zur Konfiguration des Mailservers stehen folgende Befehle zur Verfügung:

```
MAILServer +L      : Mailserverliste einrichten
MAILServer -L      : Mailserverliste entfernen
MAILServer Newlist : Mailserverliste einrichten
MAILServer Dellist : Mailserverliste entfernen
MAILServer DEScrip : Beschreibung zur Mailserverliste erstellen
MAILServer Options : Optionen zur Mailserverliste setzen
MAILServer +U      : Rufzeichen hinzufügen
MAILServer -U      : Rufzeichen löschen
MAILServer ADDUser : Rufzeichen hinzufügen
MAILServer DELUser : Rufzeichen löschen
MAILServer +M      : Administrator hinzufügen
MAILServer -M      : Administrator löschen
MAILServer ADDMain : Administrator hinzufügen
MAILServer DELMain : Administrator löschen
MAILServer Info    : Beschreibung ausgeben
MAILServer List    : Beschreibung ausgeben
MAILServer SUbscr  : Nachrichten abbonieren
MAILServer Unsubs  : Nachrichten abbestellen
MAILServer Reset   : Zähler zurücksetzen auf 1
MAILServer Setnum  : Zähler setzen
```

Mit dem Sysopbefehl MAILLISTSERV kann der Mailserver aktiviert/deaktiviert werden:

```
maillistserv 0   inaktiv
maillistserv 1   aktiv
maillistserv 2   aktiv, nur der Boxsysop kann Mailserverlisten einrichten
```

Achtung: Der Mailserver ist nur dann in der Mailbox verfügbar, wenn er als Option beim Kompilieren der Mailbox aktiviert wurde!

16.3. Einrichtung externer automatischer Server

Verschiedene Funktionen (z.B. das Einlesen der aktuellen Wetterwerte von einer Wetterstation) können regelmäßig von der Box durchgeführt werden. Dazu stehen periodisch aufgerufene Skript-Dateien zur Verfügung. Der Name und der Zeitpunkt des Aufrufs des Batch- bzw. Import-Dateien wird durch die Datei *crontab.bcm* gesteuert.

Es werden jeweils 2 Dateien mit der Endung *.bat* und *.imp* ausgeführt. Die *.bat*-Datei wird dabei dem Kommandointerpreter des jeweiligen Betriebssystems übergeben. Die *.imp*-Datei wird innerhalb der Box ausgeführt, d.h. in dieser Datei können Boxbefehle wie "send" oder "ps" enthalten sein.

Dabei wird immer zuerst die BATch-Datei ausgeführt und anschließend die dazugehörige IMPort-Datei. Existiert eine in *crontab.bcm* angeführte *.bat*- oder *.imp*-Datei nicht, so hat das keine Folgen. Es kann zu einem Eintrag in *crontab.bcm* auch nur eine *.bat*- oder *.imp*-Datei existieren, oder keines der beiden. Während der Ausführung des BATCH-Jobs "steht" die DOS-Box, es ist darauf zu achten, dass es durch oft aufgerufene DOS-Batch-Jobs nicht zu allzu großen Verzögerungen im Box-Betrieb kommt, und dass ein einzelner DOS-BATCH-Job nicht länger als 30 min dauert. Unter Linux und Windows besteht diese Zeitbegrenzung nicht, allerdings ist hier darauf zu achten, dass mehrere Batches eventuell gleichzeitig aufgerufen werden.

Bei der Abarbeitung der Import-Dateien wird ein Login mit dem Boxnamen mit Sysopstatus durchgeführt. Es können also alle Befehle ausgeführt werden, die ein Sysop ausführen darf. Zum Speichern einer Nachricht unter einem bestimmten Rufzeichen ist also z.B. folgendes zu schreiben:

```
call dl2ja
e fsg_wx 1-
s fsg_wx @ db0fsg neues Wetter
..blabla..
nnnn
impdel halfhour.imp
```

Denkbar ist, dass von *halfhour.bat* (oder welchem **.bat* auch immer) eine Import-Datei erzeugt wird, die dann anschließend eingelesen wird. Innerhalb eines Import-Vorgangs ist sowohl ein Export möglich, als auch ein verschachtelter Import. Damit ist es auch möglich, dass eine Import-Datei durch OSHELL-Befehle im Import-Prozess erzeugt wird, und anschließend im selben Import-Prozess noch importiert wird.

Unbedingt zu beachten ist, dass ein Import-Prozess, der innerhalb eines anderen Import-Prozesses erzeugt wird, gleichzeitig zum Mutterprozess abläuft. Ist der Mutterprozess also von einem Ergebnis des Tochterprozesses abhängig, so muss diesem durch ein SLEEP-Kommando Folge getragen werden. Außerdem ist das Prozess-System der OpenBCM-Mailbox flach, d.h. wenn der Mutterprozess beendet wird, läuft der Tochterprozess evtl. noch weiter.

Zum Testen von Batch- und Import-Dateien dient der Befehl BATCH. Mit diesem kann der Vorgang angestoßen werden. Die Endung der Dateien wird nicht angegeben:

batch save **führt *save.bat* und *save.imp* aus**
batch hour17 **führt *hour17.bat* und *hour17.imp* aus**

Die Ausgaben der Batch-Dateien werden unter DOS in gleichnamige Dateien mit der Erweiterung *.OUT* bzw. *.ERR* umgeleitet. Die Ausgabe der Batch-Datei *hour.bat* erfolgt also z.B. in die Dateien *hour.out* und *hour.err*.

17. Mail-Bake

In der Mail-Bake werden alle Rufzeichen gesendet, für deren Inhaber eine Nachricht vorhanden ist, die von ihm noch nicht gelesen wurde. Das Rufzeichen wird dabei nicht in die Liste aufgenommen, wenn der Benutzer nach dem Eintreffen der neuen Nachricht bereits eingeloggt war. Außerdem werden nur Mails berücksichtigt, die nicht älter sind als mit dem Parameter OLDESTBEACON in der Datei *init.bcm* angegeben (die Angabe erfolgt in Tagen). Die Aussendung eines Rufzeichens erfolgt unabhängig davon, ob es sich um die Heimatmailbox des Benutzers handelt.

Sind ungelesene Nachrichten für mehr User vorhanden, als in einer Bake Rufzeichen untergebracht werden können, so werden mehrere Mail-Baken generiert.

Die Zeitpunkte der Ausstrahlung der Mail-Bake werden durch die Datei *crontab.bcm* eingestellt. Soll die Mail-Bake alle 20 Minuten ausgesendet werden, so ist folgender Eintrag sinnvoll:

```
[...]
   0,20,40     *     *     *     *     beacon
[...]
```

Zur Konfiguration der Dateien *beacon.bcm* und *beachead.bcm* siehe auch das Kapitel "5.9. Bakeneinstellungen (*beacon.bcm* und *beachead.bcm*)".

Über den Sysopbefehl MAILBEACON kann man die Mailbaken-Ausstrahlung auch dann erzwingen, wenn keine neuen Usermail vorliegen. Dies sollte im Regelfall aber keinen Sinn machen!

18. Externe Programme

Von der Mailbox aus können externe Programme aufgerufen werden. Bestimmte Programme können für Benutzer freigegeben werden, andere sind nur mit Sysopstatus auszuführen.

18.1. Ausführung externer Programme durch einen Sysop

Der Aufruf von einem beliebigen externen Programm erfolgt für den Sysop durch die Eingabe von

```
   OSHELL <Programm-Name> <Parameter>
```

am Mailbox-Prompt.

18.2. Ausführung externer Programme durch einen User

Um ein Programm für Benutzer zugänglich zu machen, muss das Programm in einer Datei namens *runutil.bcm* eingetragen sein. Siehe hierzu auch das Kapitel "5.13. Runutils (*runutil.bcm*)".

18.3. Anforderungen an externe Programme

Das aufgerufene Programm muss folgende Eigenschaften haben:

Unter DOS:

- Schnelle Ausführung! Währenddessen steht die Box für alle User! Ein Run-Utility sollte auf keinen Fall länger als etwa eine Sekunde zur Ausführung benötigen.
- Ausgaben nur über DOS/BIOS, nicht direkt in den Bildschirmspeicher.
- Keine Eingaben! Wenn das Programm auf eine Eingabe wartet, steht alles still.

- Geringer Speicherbedarf. Es steht i.A. nur wenig freier Speicher zur Verfügung (typisch ca. 100 kB, siehe Anzeige im V-Befehl "Heap")
- Es kann sich um eine COM-, EXE- oder BAT-Datei handeln.
- Der Aufruf muss "wasserdicht" sein, d.h. der User darf mit einer x-beliebigen Kommandozeile keinen Schaden anrichten können.

Unter Linux:

- Die Box läuft während der Ausführung des Run-Utilities weiter.
- Eingaben an Run-Utilities werden akzeptiert.
- Gleichzeitige Ausführung eines Run-Utilities durch mehrere User ist möglich.
- Speicherbedarf unkritisch.
- Der Aufruf muss "wasserdicht" sein, d.h. der User darf mit einer x-beliebigen Kommandozeile keinen Schaden anrichten können.

Unter Windows:

- Die Box läuft während der Ausführung des Run-Utilities weiter.
- Ausgaben nur über DOS/BIOS, nicht direkt in den Bildschirmspeicher.
- Es kann sich um eine COM-, EXE- oder BAT-Datei handeln.
- Gleichzeitige Ausführung eines Run-Utilities durch mehrere User ist möglich.
- Der Aufruf muss "wasserdicht" sein, d.h. der User darf mit einer x-beliebigen Kommandozeile keinen Schaden anrichten können.

18.4. Übergabe der Parameter an das Run-Utility

Unter DOS und Windows werden die angegebenen Parameter bis auf die Zeichen "<", ">" und "|" transparent an das Run-Utility übergeben. Ist eines der genannten Zeichen enthalten, wird der Parameter-String an dieser Stelle abgeschnitten.

Unter Linux wird allen Zeichen ein Backslash ("\") vorangestellt, so dass von der Shell kein Zeichen anderweitig interpretiert wird. Auf diese Weise können auch die Zeichen "<", ">" und "|" übergeben werden. Letzteres geschieht aus Sicherheitsgründen aber nur, wenn dies in *runutil.bcm* mit dem Parameter "-T" eingestellt wurde.

Auf jeden Fall ist es einfacher, wenn ein Run-Utility die in *rundat.bcm* unter "cmdline" (siehe unten) angegebene Kommandozeile auswertet. Hier sind sogar mehrere Leerzeichen zwischen zwei Parametern sichtbar. Allerdings muss natürlich auch das Run-Utility aufpassen, wenn es mit den Parametern weitere Shell-Aufrufe macht.

18.5. Aufbau der Datei *rundat.bcm*

Die Datei *rundat.bcm* wird von der OpenBCM-Mailbox vor jedem Run-Utility-Aufruf erzeugt, außer es wurde die Option "-Q" in *runutil.bcm* angegeben.

Achtung: Unter Linux/Windows ist diese Datei nur unmittelbar nach dem Aufruf gültig. Zu einem beliebigen späteren Zeitpunkt kann diese Datei bereits die Daten für einen anderen Run-Utility-Aufruf enthalten.

In der Datei *rundat.bcm* sind einige Informationen über den Benutzer gespeichert, der das Run-Utility aufgerufen hat. Das externe Programm kann diese Informationen entsprechend auswerten.

Die erste Zeile enthält einige Angaben zum Benutzer in folgender Reihenfolge:

- Call ohne SSID

- MYBBS
- Name
- Mit "ALTER SPEECH" eingestellte Sprache
- 1, wenn der Login mit Passwort erfolgt ist, sonst 0
- 1, wenn der Benutzer Sysop-Status hat, sonst 0
- Status (siehe ALTER STATUS)
- Helplevel (siehe ALTER HELPLEVEL)
- Einstiegsdigi (nur richtig, wenn der Connect ausschließlich über Flexnet-Digis erfolgt ist)

Diese Informationen sind nur durch Leerzeichen getrennt und müssen anhand der Reihenfolge ausgewertet werden. Zu beachten ist, dass die Informationen in der ersten Zeile nur aus Kompatibilitäts-Gründen vorhanden sind. Sie sollte nicht mehr verwendet werden, weil diese erste Zeile in späteren Versionen der OpenBCM-Mailbox evtl. nicht mehr erzeugt wird.

Ab der zweiten Zeile folgen Informationen in der Form "Schlüsselwort Wert". Die Schlüsselwörter haben folgende Bedeutung:

call
Call ohne SSID
mybbs
MyBBS (ALTER FORWARD)
name
Name (ALTER NAME)
language
Sprache (ALTER SPEECH)
password
1, wenn der Login mit Passwort erfolgt ist, sonst 0
sysop
1, wenn der Benutzer Sysop-Status hat, sonst 0
status
User-Status (ALTER STATUS)
helplevel
Help-Level (ALTER HELPLEVEL)
digi
Einstiegsdigi (nur richtig, wenn der Connect ausschließlich über Flexnet-Digis erfolgt ist)
txt_output
Datei, die nach Beendigung des Run-Utilities als Text ausgegeben wird
bin_output
Datei, die nach Beendigung des Run-Utilities binär - allerdings ohne AutoBIN-Header - ausgegeben wird
abinoutput
Datei, die nach Beendigung des Run-Utilities im AutoBIN-Format ausgegeben wird
yappoutput
Datei, die nach Beendigung des Run-Utilities im YAPP-Protokoll ausgegeben wird
imp_output
Datei, deren Inhalt nach Beendigung des Run-Utilities im Hintergrund ausgeführt wird; die Abarbeitung der Befehle geschieht mit Sysop-Status und als Prozess-Rufzeichen wird das der Mailbox verwendet. Sind die letzteren beiden Eigenschaften nicht gewünscht, so kann

- das Rufzeichen in der IMPORT-Datei mit "call <rufzeichen>" geändert werden und
- der Sysopstatus mit "PW OFF" verlassen werden.

Soll diese Datei nach der Ausführung gelöscht werden, so ist an beliebiger Stelle der Befehl "IMPDEL" ohne Parameter einzufügen.
user_imp

Datei, deren Inhalt nach Beendigung des Run-Utilities so abgearbeitet wird, als wären es Benutzereingaben gewesen

cmd_line
Befehlszeile des Benutzers, mit der der Aufruf des Run-Utilities erfolgte

login_time
Loginzeit im Unix-Format (Sekunden seit 1.1.1970)

board
Aktuelles Board

fspath
Aktueller Pfad im Filesurf

Die Auswertung der Informationen sollte unbedingt anhand der Schlüsselwörter erfolgen, und nicht anhand der Reihenfolge. Die Dateinamen bei den Schlüsselwörtern txt_output, bin_output, abinoutput, yappoutput, imp_output und user_imp sind eindeutige, temporäre Namen, die von der Box vorgegeben werden. Das Run-Utility sollte hier auch unbedingt längere Pfadnamen akzeptieren, damit künftig problemlos Änderungen auf Boxseite möglich sind.

Achtung:
Auch unter DOS und Windows werden als Pfadtrenner Slashes ("/"), und keine Backslahes ("\") verwendet! Mit Ausnahme von "imp_output" werden alle Dateien nach der Ausgabe bzw. nach der Bearbeitung automatisch gelöscht.

Beispiel einer *rundat.bcm*:

```
DGT274 DBO274.#NRW.DEU.EU Markus DL 0 0 0 0 DNX274
call       DGT274
mybbs      DBO274.#NRW.DEU.EU
name       Markus
language   DL
password   0
sysop      0
status     0
helplevel  0
digi       DNX274
txt_output temp/cctfci0
bin_output temp/cctfci1
imp_output temp/cctfci4
user_imp   temp/cctfci5
cmdline    7m
login_time 1041175316
board      DGT274
abinoutput temp/cctfci2
yappoutput temp/cctfci3
fspath     /filesurf/openbcm/beta/
```

Die Liste kann ggf. nach unten erweitert werden.

Nach Beendigung des Run-Utilities erfolgt die Bearbeitung der vom Run-Utility erzeugten Dateien und Ausgaben in folgender Reihenfolge (falls eine Datei nicht erzeugt wurde, geschieht einfach gar nichts):

- Ausführung von imp_output (erfolgt im Hintergrund, eigene Task)
- Ausgabe der STDOUT-Ausgaben des Run-Utilities
- Ausgabe der STDERR-Ausgaben des Run-Utilities
- Ausgabe von txt_output als normaler ASCII-Text
- Binäre Ausgabe von bin_output (transparent, d.h. ohne jeden Vor- und Nachspann)
- Ausgabe von abinoutput im AutoBIN-Format
- Ausgabe von yappoutput im YAPP-Protokoll
- Ausführung der Befehle in der Datei user_imp

Wie an den vielen Dateizugriffen zu sehen ist, ist für einen befriedigend schnellen Betrieb von Utilities ein ordentlicher Rechner notwendig.

19. Nicht-AX.25-Zugänge zur Mailbox

19.1. Ansteuerung einer seriellen Schnittelle unter DOS

Diese Option ist normalerweise unter DOS wegen Speichermangels nicht mehr verfügbar.

Die Box wird bei aktivierter Option über einen seriellen Port angesprochen. Dabei wird ein primitives Terminal (z.B. VT100 oder XTALK) benötigt und man erhält eine Benutzeroberfläche wie über Funk. Zum Login ist allerdings eine Anmeldung mit Rufzeichen und Passwort erforderlich. Das Passwort kann nur von einem Sysop mit

SETUSER <Rufzeichen> TTYPW <Passwort> (max. 7 Zeichen)

eingegeben werden und ist für jeden Benutzer einzeln zu vergeben. Ohne vorher eingetragenes Passwort kann man sich nicht über die serielle Schnittstelle einloggen. Dies ermöglicht, dass auch Unbefugte Zutritt zum Terminal haben können, ohne die Datensicherheit der Box zu gefährden. Beim Passwort wird Groß- und Kleinschreibung unterschieden. Durch die Identifizierung mit dem TTYPW ist man nicht automatisch Sysop der Mailbox, dazu ist eine zusätzliche Sysop-Passwort-Identifizierung notwendig.

Zur Konfiguration ist mit TTYMODE folgender Parameter einzustellen(DOS):

```
TTYMODE 1:9600,n,8,1,e
                     ^e=echo ein, l=echo lokal
                   ^Anzahl der Stopbits
                 ^Anzahl der Datenbits
               ^Parity n=no e=even o=odd
          ^baudrate 1200-38400
        ^com-Schnittstelle COM1-COM4
```

Standard-COM-Adresse/-Interrupts:

- COM1: IRQ 4, Base-Addr 0x3f8
- COM2: IRQ 3, Base-Addr 0x2f8
- COM3: z.B. IRQ 5 (nicht standardisiert), Base-Addr 0x3e8
- COM4: z.B. IRQ 7 (nicht standardisiert), Base-Addr 0x2e8

Ist ein 16550 (Baustein mit Puffer) eingebaut, so wird dieser erkannt und benutzt. Als Handshake wird stets XON/XOFF verwendet. Es sind deshalb neben den Datenleitungen (GND,TXD,RXD) auch die Handshakeleitungen zu verbinden.

Bei der Anwendung dieser Schnittstelle ist der Phantasie freier Lauf gelassen. Denkbar wäre z.B. eine Anbindung an ein Modem. Das Abschalten des TTY-Ports erfolgt mit TTYMODE OFF (Default).

Wichtig: Die Installation des TTY-Ports wird erst beim nächsten Start der DOS-OpenBCM-Mailbox durchgeführt.

19.2. Zugang über TELNET unter Linux und Windows

Die Mailbox kann unter Linux/Windows über Telnet angesprochen werden.

Diese Funktion macht aufgrund der hohen Laufzeiten über Funk wenig Sinn. Jedoch innerhalb eines drahtgebundenen Netzes stellt dies die beste Möglichkeit dar, die OpenBCM-Mailbox zu erreichen.

Die OpenBCM-Mailbox erwartet normalerweise auf dem TCP Port 4719 einen Telnet Verbindungsaufbau. Dieser Port kann in *init.bcm* (Parameter TELNET_PORT) geändert werden. Die Erreichbarkeit über Telnet kann auch durch Setzen dieses Wertes auf 0 ganz abgeschaltet werden.

Folgen wir zur Tat:

```
bash$ telnet server 4719
Trying 172.16.1.1...
Connected to server.local.net.
Escape character is '^]'.

OpenBCM-Mailbox v1.05 (Linux)
DH8YMB login: dh8ymb
password: xxxxxxx

OpenBCM-Mailbox v1.05 (Linux) - Silly Valley - 01:10z
Hallo Markus, Helplevel=2, Zeilen=0, letzter Login 24.07.02 12:12z

Inhaltsverzeichnis für DH8YMB @ DH8YMB:
372) DH8YMB 24.11.02 11:03 13 #999 @ DH8YMB OpenBCM Demo
(12:13 DH8YMB)-->
```

Ob ein Passwort und welches Passwort eingegeben werden muss, hängt dabei von folgenden Aspekten ab:

- *rhosts.bcm* - In dieser Datei werden alle 'trusted hosts' eingetragen, damit sind die Namen oder IP-Adressen von Rechnern gemeint, denen vertraut werden kann und die dann gleich automatisch Sysopstatus besitzen, siehe hierzu auch Kapitel "5.7. Remotehost (*rhosts.bcm*)".
- Bei allen anderen Logins kann das Passwort nur durch den Sysop gesetzt werden. Es gilt dabei der Parameter (A TTY). Nach der korrekten Eingabe des Passwortes ist der Benutzer eingeloggt und besitzt keinen Sysopstatus.

Das Auslesen von Binärmails ist im Allgemeinen nicht besonders sinnvoll.

19.3. Zugang über HTTP unter Linux und Windows

Im Gegensatz zu den meisten anderen Lösungen (CGI-Skript), besitzt die OpenBCM-Mailbox ein eingebautes HTTP Interface.

Dadurch lässt sich die Box voll interaktiv bedienen. Das bedeutet Befehle werden direkt ausgeführt und Benutzer können sich unter ihrem eigenen Rufzeichen einloggen.

Der HTTP Server ist normalerweise auf dem TCP Port 8080 erreichbar. Andere Werte lassen sich über *init.bcm* (Parameter HTTP_PORT) setzen. Die Erreichbarkeit über HTTP kann auch durch Setzen dieses Wertes auf 0 ganz abgeschaltet werden.

Bei der ersten HTTP Verbindung zu einer OpenBCM-Mailbox kommt ein kleines Fenster auf den Bildschirm. Dort wird man aufgefordert einen Namen und ein Passwort einzugeben. Auch hier gelten ein paar Konventionen bezüglich der Eingaben: Als Name wird grundsätzlich das eigene Rufzeichen eingegeben. Beim Passwort trifft die OpenBCM-Mailbox eine kleine, aber wichtige Unterscheidung:

- Wird eine Verbindung vom localhost [127.0.0.1] oder dem Packet Radio Netz (44.x.x.x.) aufgebaut, dann gilt als Passwort der in der OpenBCM-Mailbox eingegebene Name des Benutzers, sofern in *init.bcm* nicht HTTPTTYPW auf 1 gesetzt wurde. Durch diese Konvention muss den Benutzern durch den Sysop kein Passwort eingetragen werden. Hat ein Benutzer ein AX.25-Password gesetzt, ist kein Login möglich, es sei denn, der Benutzer hat in dem AX25-Passwort-String die Buchstabenkette "DUMMY" mit eingefügt, dann wird hier ebenfalls der Name des Benutzers abgefragt. Ist HTTPTTYPW auf 1 gesetzt, so wird generell nur das Passwort, das unter ALTER TTYPW gesetzt wurde, akzeptiert.
- Bei Verbindungen aus anderen Netzen gilt immer grundsätzlich das TTYPW als Passwort.

- Der Sysop hat die Möglichkeit, durch Definieren eines GUESTCALL und Setzen der Variable HTTPGUESTFIRST auf den Wert 1 in *init.bcm*, den Login-Vorgang dahin abzuändern, dass generell bei HTTP-Zugriff der Login unter dem GUESTCALL erfolgt, wobei hier dann nur Lesezugriff gegeben ist.
- Das händische Generieren eines TTYPW für jeden User durch den Sysop kann umgangen werden, indem die Variable HTTPACCOUNT in *init.bcm* auf den Wert 1 gesetzt wird. Dann hat ein User die Möglichkeit sich unter dem GUESTCALL einzuloggen, und sich für sein Rufzeichen selber ein TTYPW zu setzen. Hierbei verbirgt sich jedoch das Risiko, dass jemand einem anderen ein TTYPW setzt, und daher sollte diese Funktion eigentlich nicht angewendet werden! Ich rate vom Setzen von HTTPACCOUNT auf den Wert 1 also dringend ab!

Nach einem erfolgreichen Login setzt der HTTP Server beim Benutzer ein Cookie. Das Cookie dient dazu, bei einem späteren Login den Benutzer zu identifizieren und dadurch die Passwortabfrage zu umgehen. Wird dem Browser verboten Cookies zu setzen, dann muss das Passwort jedes Mal auf das Neue eingetippt werden. Als Browser eignen sich in jedem Fall die Produkte von Netscape bzw. Microsoft, alle anderen Browsern müssen Frames unterstützen. Die Funktionen der OpenBCM-Mailbox lassen sich komplett Fernbedienen, jedoch gibt es einige Ausnahmen:

- Beim Senden von Nachrichten muss immer die Referenz zu send angeklickt werden. Es hilft nichts, in der Befehlszeile einen Send-Befehl zu schreiben und dann auf Execute zu klicken.
- Die Bedienung von interaktiven Run-Utilities ist nicht möglich.
- Beim Empfang von Nachrichten werden Binärmails automatisch umgesetzt. Dabei wird der Mimetype beachtet und folgerichtig werden beispielsweise Mails mit einem binär eingespielten JPG-Bild automatisch angezeigt.

Alle anderen Funktionen sollten sich wie gewohnt bedienen lassen.

In einem Verzeichnis */bcm/http* lassen sich übrigens Dateien ablegen, die dann auch über HTTP abrufbar sind. Allerdings funktionieren nur die Endungen *.htm* (bzw. *.html*), *.gif*, *.wav* und *.jpg*.

Wird in */bcm/http* eine Datei *back.jpg* abgelegt, so wird dies als Hintergrundbild eingeblendet, ist eine Datei *qsl.wav* dort abgelegt, so wird diese bei jedem Aufruf ausgegeben.

19.4. Zugang über SMTP und POP3 unter Linux und Windows

Die OpenBCM-Mailbox unterstützt ein TCP/IP taugliches Mailinterface.

Das bedeutet: Emails können via SMTP in das Packet Radio Netz eingespielt werden, dabei werden die Mails automatisch umgesetzt und an die Packet Radio Verhältnisse angepasst. Auch die umgekehrte Richtung, nämlich das Abrufen der eigenen Mails via POP3 ist möglich. Auch hier erfolgt eine automatische Anpassung der Packet Radio Mails an das übliche Emailformat.

- Der OpenBCM-Mailbox-SMTP-Server ist auf Port 8025 erreichbar. Andere Werte lassen sich über *init.bcm* (Parameter SMTP_PORT) setzen. Die Erreichbarkeit über SMTP kann auch durch Setzen dieses Wertes auf 0 ganz abgeschaltet werden. Beim Versenden von Nachrichten ist es wichtig, die Absenderadresse im Mailprogramm korrekt einzutragen, also z.B. "dg9mhz@db0aab.#bay.deu.eu". Als mögliche Adressaten kommen Benutzer mit gültigen Rufzeichen oder Rubriknamen in Frage. Vom SMTP Server werden pro Mail bis zu 100 verschiedene Empfänger unterstützt, somit ist SMTP bei der OpenBCM-Mailbox ideal für die Verteilung von Mailinglisten.

- Zum Abrufen der eigenen Mails dient der OpenBCM-Mailbox-POP3 Server auf Port 8110. Wie üblich lassen sich andere Werte in *init.bcm* setzen.

Das Abrufen von Mails erfordert die Angabe eines Benutzernamens und Passwortes. Hier gelten die gleichen Konventionen, wie beim HTTP-Server.

Durch die automatische Umsetzung der Packet Radio Mails in Emails, verlieren die Mails keinerlei Information. Binärdateien und 7Plus Einspielungen werden automatisch Base64 kodiert. Ebenso automatisch wird der Mimetype gesetzt. Über SMTP eingespielte Mails bleiben von diesen Algorithmen unberührt und werden dadurch transparent durchgereicht.

Ein Konfigurationsbeispiel für db0aab anhand des Netscape Communicator's V4.01:

Zunächst wird über den mitgelieferten 'User Profile Manager' ein neues Profil angelegt. Die nun folgenden Formulare werden korrekt ausgefüllt:

Im Feld Namen bitte den vollständigen Namen eingeben, z.B. "Deti Fliegl". Als Emailadresse ist die komplette Adresse einzugeben, also z.B. "dg9mhz@db0aab.#bay.deu.eu". Das Profil nennt man dann z.B. "OpenBCM".

Nun gibt man noch mal Namen und Emailadresse ein. In der Zeile SMTP Server trägt man ein "db0aab.ampr.org:8025". Der Mailserver Username ist in dem Beispiel "dg9mhz". Als Incoming Mailserver gibt man an "db0aab.ampr.org:8110".

Der Communicator wird neu gestartet und nun wählt man unter "Edit/Preferences" die Kategorie "Mail&Groups/Mail Server" aus. In der Zeile unter POP3 gibt es eine Checkbox zum Anklicken, neben der steht: "Leave Messages on server after retrieval". Diese sollte angeklickt sein, damit die Packet Radio Mails auch nach der Abholung über POP3 weiterhin in der OpenBCM-Mailbox verfügbar sind. Wer das nicht will, kann die Option auch ausgeschaltet lassen.

Hier endet das allgemeine Konfigurationsbeispiel. Generell gilt: Wenn der TCP/IP Server keinen Nameserver besitzt (DNS), dann müssen entweder jeweils die IP Adressen an Stelle der Domain Namen direkt eingetragen werden oder man trägt in der Datei *hosts* des Betriebssystems die IP Adressen und Namen per Hand ein:

- Bei Windows findet sich die Datei unter *\windows\hosts*
- Bei Windows NT/2000/XP unter *\winnt\system32\drivers\etc\hosts*
- Bei Unix/Linux unter */etc/hosts*

19.5. Zugang über NNTP unter Linux und Windows

Der NNTP Server der OpenBCM-Mailbox befindet sich momentan (Januar 2004) noch in einem Betastatus. Der NNTP-Server ist auf Port 8119 erreichbar. Andere Werte lassen sich über *init.bcm* mit dem Parameter NNTP_PORT setzen. Die Erreichbarkeit über NNTP kann auch durch Setzen dieses Wertes auf 0 ganz abgeschaltet werden.

Es sind noch längst nicht alle NNTP-Befehle implementiert. So kann man momentan auch nur Mails über NNTP lesen, jedoch keine versenden.

Es ist auch nicht auszuschließen, dass es mit dem einen oder anderen Mailclient noch Probleme gibt. Gut funktionieren tut aber mittlerweile der Newsreader, der in Netscape implementiert ist. Auch mit Outlook oder TheBat kann man Mails problemlos abrufen und lesen.

19.6. Zugang über FTP unter Linux und Windows

Der FTP Server der OpenBCM-Mailbox ist die jüngste Servererweiterung, scheint aber schon recht stabil zu laufen. Der FTP-Server ist auf Port 8023 erreichbar.

Andere Werte lassen sich über *init.bcm* mit dem Parameter FTP_PORT setzen. Die Erreichbarkeit über FTP kann auch durch Setzen dieses Wertes auf 0 ganz abgeschaltet werden.

Alle Dateien, die im Filesurf angeboten werden, sind über den FTP ebenfalls downloadbar - es gilt hier also auch die Einstellung von FSPATH in *init.bcm*. Ein Upload ist nur in die bei FSPATH mit Schreibzugriff festgelegten Verzeichnisse möglich. Ein Löschen oder Umbenennen von Dateien ist generell nicht möglich.

19.7. Zugang per Service-Interface unter Linux und Windows

Es existiert ein simples Service-Interface außerhalb der normalen Box-Oberfläche, das über Telnet erreichbar ist und auch darüber bedient werden kann.

Hier sind insgesamt vier Befehle möglich:

- TRACE ON/OFF	Startet/Stoppt die Ausgabe von *trace/syslog_r.bcm*
- MONITOR ON/OFF	Monitorfunktion an/aus
- CMD <Befehl>	Führt einen Boxbefehl aus
- QUIT	Verlässt das Service-Interface

Das Interface funktioniert nur dann, wenn der Host, auf dem das Telnet gestartet wird, in der Datei *rhosts.bcm* eingetragen ist und der Serv-Port entsprechend definiert wurde (normalerweise Port 8123).

19.8. Die Net-CMD-Schnittstelle unter Linux

Die OpenBCM bietet ein Radio-Inferface (Net-CMD-Interface) unter Linux, wie es Wampes bzw. das Terminalprogramm TNT unterstützt, über dessen TCP/IP-Socket man Verbindungen nach außen aufbauen kann. Dazu muss zunächst mit dem Befehl RADIO_PORT die Portnummer des Radio-Interface definiert werden (standardmäßig auf 0 gesetzt, und damit deaktiviert).

Beispiel:
Radio-Port setzen:
radio_port 8134
Eine Telnet-Verbindung aus dem System mit
telnet localhost 8134
verbindet mit dem Radio-Interface, dort kann dann mit dem Befehl
CONNECT AX25 OE1XLR OE3DZW-12
als "oe3dzw-12" nach "oe1xlr" eine Verbindung aufgebaut werden.

Allerdings unterscheiden sich die Interfaces von Wampes und TNT, die Implementierung hat sich bei OpenBCM an TNT orientiert. Das Net-CMD-Interface ist aber kompatibel mit Wampes.

Nach einem Connect zu dem Socket, ist das Interface im Kommandomodus und akzeptiert drei Befehle: ASCII, BINARY und CONNECT. Eine Eingabe von anderen Befehlen oder falschen Parameter zu den drei Befehlen führt zu einem Beenden der Verbindung.

ASCII wählt eine Zeichenumwandlung von LF nach CR vor dem Senden von Daten auf der AX25-Seite und umgekehrt aus. Dies ist die Normaleinstellung.

BINARY wählt eine transparente Verbindung ohne Zeichenumwandlung aus.

CONNECT startet einen AX25-Verbindungaufbau. Es werden weitere Parameter benötigt, der genaue Syntax ist:
CONNECT <transport mode> <destination callsign> [source callsign]

Der einzig gültige Wert für <transport mode> ist AX25, andere Modi führen zu einem Beenden der Verbindung. Das <destination callsign> darf keine Digipeater beinhalten.

Nach einem erfolgreichen Verbindungsaufbau wechselt das Interface in den Datamodus: alle empfangene Daten werden an das Socket gesendet, alle Daten vom Socket werden auf der AX25-Seite übertragen.

Wenn der Verbindungsaufbau nicht erfolgreich war, wird die Interfaceverbindung ohne weitere Infos beendet.

Beim Verbindungsaufbau werden die Pfade entsprechend einer Datei *netpath.bcm* verwendet:

```
------ l2path.bcm
; comments are ignored (start with ;)
; format: <to_call>: <path>
db0clx:  db0clx oe1xlr
ha5dxx:  ha5dxc oe1xlr oe3xpr
------
```

20. Mehrsprachigkeit der Mailbox

Die OpenBCM-Mailbox in der Lage, bis zu 40 verschiedene Sprachen zu sprechen, derzeit sind folgende Sprachen vorhanden:

- DL: Deutsch
- BAD: Badisch
- BAY: Bayerisch
- BW: Schwäbisch
- CT: Portugiesisch
- EA: Spanisch
- FF: Französisch
- GB: Englisch
- HA: Ungarisch
- HRV: Kroatisch
- I: Italienisch
- KL: Kölsch
- LX: Letzeburgisch
- NL: Holländisch
- OK: Tschechisch
- OM: Slowakisch
- PF: Pfälzisch
- PL: Polnisch
- RUS: Russisch
- S5: Slowenisch
- TA: Türkisch
- TRK: Türkisch mit Akronymen

Hinweise:

- Alle Sysopmeldungen sind Englisch. Um den Aufwand vertretbar zu halten, sind die Meldungen am Bildschirm und die für die reinen Sysop-Befehle nicht änderbar und permanent in englischer Sprache eingebaut. Da es sich aber meist um einfache Ausdrücke handelt, dürften keine Probleme zu erwarten sein.
- Änderungen an der Boxsoftware, insbesondere Erweiterungen der Funktionalität bedingen gelegentlich eine Erweiterung der Dateien mit den Meldungen. Das erschwert den Updatevorgang erheblich, besonders wenn sehr viele Sprachen existieren. Seit der ersten Stunde der Mehrsprachigkeit ist bis jetzt das Format der messages-Datei mehrfach

angepasst worden. Es ist vorherzusehen, dass in einer der nächsten Versionen nochmals ein Update erforderlich wird, weil schon recht viele Meldungen in der Box festverdrahtet in englischer Sprache eingebaut sind.

- Die Meldungs-Dateien müssen im Format exakt stimmen. Eine Zeile zu viel oder zu wenig kann unvorhersehbare Folgen haben. Eine gewissenhafte Pflege dieser Dateien ist deshalb wichtig.
- Wird eine Datei zu einer angewählten Sprache nicht gefunden, so wird zunächst nach einem englischen Text (Endung *.GB*) gesucht, und wenn auch dieser nicht vorhanden ist wird die Meldung in Deutsch (*.DL*) ausgegeben. Einer der beiden Meldungsblöcke *.GB* oder *.DL* muss existieren, sonst erfolgt ein Abbruch der Box. Analoges gilt für die Dateien INFO, AKTUELL, HELP, QTEXT, etc., allerdings ist deren Existenz nicht lebenswichtig.

Erzeugung einer neuen Sprache:
Um eine neue Sprache zu erzeugen, muss in erster Linie eine Datei *msg/messages.<Sprachkenner>* erzeugt werden. Der <Sprachkenner> ist dabei frei wählbar, sollte aber möglichst den ersten zwei Buchstaben des Calls eines Landes in dem diese Sprache gesprochen wird entsprechen. Vordefiniert sind *.DL* für Deutsch und *.GB* für Englisch. Sprachen denen kein Landeskenner zugeordnet werden kann (weil es keine entsprechenden Rufzeichen gibt) sollten sinnvollerweise 3stellige Kennungen (z.B. *.BAY* für Bayerisch) haben, dann werden sie nur manuell vergeben.

20.1. Aufbau einer Sprach-Datei

Die erste Zeile einer Sprach-Datei enthält eine Beschreibung der Sprache und eine Versionsnummer. Beides kann man beim Aufruf von "A S" abrufen.

Die Landeskenner werden in der Datei *speech.bcm* verwaltet, siehe Kapitel "5.8. Sprachdefinitionsdatei (*speech.bcm*)".

Ab der zweiten Zeile kommen die eigentlichen Meldungen. Sie sind in einer C-kompatiblen Form abgelegt. Zu beachten ist folgendes:

- Jede Zeile beginnt und endet mit einem Anführungs-Zeichen. Beginnt eine Zeile mit ";", so wird sie ignoriert. Zeichen nach dem zweiten Anführungszeichen werden ebenfalls ignoriert.
- Steuerzeichen werden in C-Syntax abgelegt. Ausgewertet werden
  - \n: Return (Zeilenende)
  - \a: CTRL-G (Klingelzeichen)
  - \": Anführungszeichen
  - \\: Backslash
- Variablen, die innerhalb von den Ausgabestrings ausgegeben werden, werden als C-Formatstring abgelegt. Folgende Symbole tauchen auf:
  - %s: String
  - %c: Einzelzeichen
  - %d, %u, %ld: Dezimalzahl

  Dabei darf die Position der Ausgabefelder verändert werden, keinesfalls aber der Typ oder die Reihenfolge.
- Es gibt Ausgaben, deren Länge exakt stimmen muss (z.B. Überschriften). Dies ist meist aus dem Zusammenhang zu ersehen und sollte beachtet werden.
- Die beste Methode beim Übersetzen ist, die deutsche oder englische Datei umzubenennen, in den Editor zu nehmen und Zeile für Zeile zu übersetzen.

Die Indizes für die verschiedenen Help-Dateien (*msg/help.**) werden auf der Platte verwaltet. Dadurch entstehen zusätzliche Dateien (*msg/helpidx.**). Diese

werden bei Änderung der Help-Datei automatisch angepasst, es ergibt sich kein Handlungsbedarf für den Sysop.

Es können maximal 40 Sprachen verwaltet werden, sofern der Speicherplatz ausreicht.

20.2. Aufbau einer Help-Datei

Neben der Datei *msg/messages.** müssen auch die anderen Text-Dateien, insbesondere *msg/help.** übersetzt werden. Die Endung dieser Dateien muss dann passend zum jeweiligen Sprachkenner sein.

Bei der Datei *msg/help.<Sprachkenner>* werden Makros nicht expandiert, ein "%s" wird als solches ausgegeben. Die Help-Datei besteht aus durch Haupt- und Unterbegriffe referenzierte Texte. Die Begriffe werden mit "\\begriff" in der ersten Spalte in einer eigenen Zeile definiert.

Ein "help alter forward" kann mit "h a f" abgekürzt werden und gibt einen anderen Helptext als bei Eingabe von "help alter" aus. In der Help-Datei werden solche Kombinationen mit einem Punkt abgetrennt.

Beispiel:

```
\\ALTER
--- Hilfe-Text für den Befehl ALTER ---
\\ALTER.FORWARD
--- Hilfe-Text für den Befehl ALTER FORWARD ---
```

Es ist nur eine Hierarchieebene möglich, Konstrukte wie "\\DIR.USER.MESSAGES" funktionieren nicht.

Außerdem können für redundante Helptexte Verknüpfungen angelegt werden. Dies erfolgt mit dem Zeichen "=".

Beispiel:

```
\\MSG=TALK
--- Kein Text! ---
```

ruft bei der Eingabe "help msg" den Helptext von "talk" auf.

```
\\MYBBS=ALTER.FORWARD
--- Kein Text! ---
```

ruft bei Eingabe von "h mybbs" den Helptext von "alter forward" auf.

Auch die umgekehrte Verknüpfung wäre zulässig:

```
\\ALTER.FORWARD=MYBBS
--- Kein Text! ---
```

Es lassen sich also Unter- und Überbegriffe beliebig vermischen. Verknüpfungen dürfen wiederum auf Verknüpfungen zeigen, zyklische Verknüpfungen sind natürlich unzulässig und führen zu einer Fehlermeldung im Syslog.

Ich bitte dringend darum, jede Übersetzung an BAYCOM @ BABOX zu senden, um eine zentrale Lagerhaltung aller Texte zu ermöglichen.

21. Umstellung DOS-Version auf Windows oder Linux

21.1. Umstellung von DOS/Windows auf Linux

Der Umstieg von der DOS- oder Windows-Version auf Linux ist unkomplizierter, als man vermuten möchte! Alle vorhandenen Mails können 1:1 auf das Linux-System kopiert werden und gehen bei der Umstellung nicht verloren!

Es sind jedoch ein paar Kleinigkeiten zu beachten, die nachfolgend beschrieben werden:

Es ist vor der Umstellung zunächst zu überprüfen, welche BCM/OpenBCM-Version unter DOS/Windows eingesetzt wird.

Ab OpenBCM v1.04 verwendet die Mailbox die Userdatenbank *users4.bcm*, die auch unter Linux verwendet wird. Diese kann also unter Linux 1:1 übernommen werden, der nachfolgende Absatz kann dann also übersprungen werden.

Wird unter DOS/Windows eine ältere OpenBCM-Version oder gar noch eine BCM-Version eingesetzt, die noch die Userdatenbank *users.bcm* einsetzt, so muss diese manuell mit einem Konvertiertool umgestellt werden oder zunächst die DOS/Windows-Version auf den aktuellsten Stand gebracht werden, wobei dann die Konvertierung automatisch geschieht. Entscheidet man sich für die manuelle Konvertierung, kann diese nur unter DOS bzw. unter Windows im DOS-Fenster erfolgen, jedoch nicht unter Linux! Man benötigt für die manuelle Konvertierung das Programm *cvusers4.exe*, das einfach im DOS-BCM/OpenBCM-Verzeichnis zu starten ist. Nachdem *cvusers4.exe* erfolgreich durchgelaufen ist, sollte eine Datei *users4.bcm* entstanden sein - die Userdatenbank für die aktuelle OpenBCM-Version.

Als nächstes kopiert man nun die Datei *users4.bcm* und die Dateien *userh2.bcm*, *hadr4.bcm* und *hadrhash.bcm* auf das Linux-System ins dortige OpenBCM-Verzeichnis. Dabei ist darauf zu achten, dass die Dateinamen komplett aus Kleinbuchstaben bestehen, ggf. ist also mit dem Shell-Befehl "mv" der Dateiname unter Linux anzupassen!

Die BID-Datei ist nicht konvertierbar und muss unter Linux durch REORG F aus dem Mail-Datenbestand neu erzeugt werden.

Der nächste Schritt ist dann die Überführung aller vorhandenen Daten (also komplettes OpenBCM-Verzeichnis sowie USER und INFO-Verzeichnisbäume) von DOS/Windows nach Linux. Folgende Möglichkeiten sind denkbar:

- Benutzung eines neuen Rechners und Übertragung der Dateien von alt nach neu über Ethernet (mit NFS, Samba-SMB oder FTP)
- Einbau einer neuen Platte, Installation von Linux, mounten der alten Platte als MSDOS-Dateisystem, dann umkopieren der Daten auf die neue Platte
- Sicherung aller Dateien auf ein vorhandenes Medium, das sowohl unter DOS/Windows als auch unter Linux ansprechbar ist (z.B. CD-ROM), dann Einspielen auf die unter Linux neu formatierte Platte

Das OpenBCM-Home Verzeichnis sollte unter */bcm* angelegt werden. Hier werden alle Dateien der DOS/Windows-Version mit allen Unterverzeichnissen hineinkopiert. Selbstverständlich ist es sinnvoll, hierbei die rein DOS/Windows-spezifischen Dateien auszumisten. Die Verzeichnisse für die *user*- und *info*-Daten können in *init.bcm* frei gewählt werden, sind aber im Regelfall Unterverzeichnisse vom Hauptverzeichnis */bcm*.

Werden mehrere Platten verwendet, so müssen hier sinnvolle Mountpunkte definiert werden. Es kann nicht im Rootverzeichnis von Platten oder Partitionen gearbeitet werden, da hier Linux das lost+found-Verzeichnis anlegt, was die Mailbox durcheinander bringen kann. Es ist also notwendig, vom Mountpunkt stets ein Unterverzeichnis zu definieren. Die Pfade sollten jedoch nicht zu lang sein, da dies etwas die Performance mindert.

Sind alle Dateien umkopiert und die Pfade eingestellt, so kann die Box gestartet werden. Vor einem Zugriff über Funk sollten noch nacheinander "REORG L", "REORG I", "REORG H" und "REORG F" gemacht werden. Hiermit wird auch die BID-Datei neu aus den Mail-Datenbeständen erzeugt. Hinweis: Jeder REORG muss abgewartet werden, ehe der nächste gestartet werden kann.

Die beiden Versionen unter DOS/Windows und Linux verhalten sich bezüglich Dateisystem so identisch, wie nur möglich. Ein Fallstrick ist die Tatsache, dass bei UNIX/Linux Dateinamen üblicherweise kleingeschrieben werden, bei DOS ist GROSS-/Kleinschreibung egal und es wird dummerweise meist alles GROSS geschrieben. Das schafft Verwirrung und war auch in der Software nicht ganz einfach zu implementieren, zumal viele Dateinamen aus dem Zusammenhang automatisch erzeugt werden. Die Linux OpenBCM-Mailbox verwendet ausschließlich kleingeschriebene Dateinamen, egal wo sie auftauchen.

Es darf (fast) überall die Folge CR/LF statt der UNIX/Linux-üblichen Folge LF in den Dateien vorkommen. Umgekehrt kommt auch die DOS-Version mit nur-LF aus. Ausnahme sind *list.bcm* und *check.bcm* Dateien. Diese müssen bei beiden Versionen stets mit CR/LF abgeschlossen sein. Diese werden jedoch üblicherweise nicht mit einem Editor bearbeitet und sind daher unproblematisch.

Wichtig ist, dass nach der Installation alle Dateien (auch das Verzeichnis selbst!) dem User "bcm" gehören (siehe oben). Sicherheitshalber sollte also:

```
cd /bcm (Home-Verzeichnis der OpenBCM)
cd ..
chown -R bcm:bcm bcm
chmod -R u+rw bcm
```

aufgerufen werden.

21.2. Umstellung von DOS auf Windows

Spätestens mit der Verbreitung von Windows XP (vorher Windows NT und 2000, Windows 3.1/95/98/ME ist jedoch unbrauchbar) ist quasi fast jedem ein auch für den Mailboxbetrieb durchaus nutzbares Betriebssystem bekannt, dass im Vergleich zu DOS vor allem das lästige 640 kB Hauptspeicher-Problem verbannt und ein vernünftiges Dateisystem NTFS zur Verfügung stellt. Schon alleine dies sind gute Gründe von der DOS-Version wegzukommen. Sicherlich ist die Linux-Welt eine noch bessere Wahl, da es hier vor allem viel mehr Packet-Radio-Software gibt als unter Windows.

Die Windows-Version verhält sich sehr viel ähnlicher zur Linux- als zur DOS-Version. Das ist auch kein Wunder, da im 32 Bit-Mode vieles aus der DOS-Welt (z.B. Hardware- und BIOS-Zugriffe) nicht zur Verfügung steht, und auch z.B. die Speicherverwaltung eher vergleichbar mit Linux ist.

Der Unterschied zu DOS ist:

- Es ist keine direkte Benutzeroberfläche vorhanden. Ein Login ist nur über Telnet (Standard: TCP-Port 4719) und HTTP (Standard: TCP-Port 8080) möglich.
- Der Layer2 für Packet-Radio ist in *bcm32.exe* enthalten. Es ist allerdings nur KISS und AXIP über UDP möglich. Die Parametrierung erfolgt über *init.l2*, wie bei Linux.

Alle Datenstrukturen sind kompatibel zur DOS-Version. Es muss hier also beim Umstieg von DOS zu Windows nichts konvertiert werden.

22. Umstellung von BCM v1.42n auf OpenBCM

Die Version 1.42n war die letzte BCM-Version, die von OE3DZW herausgegeben wurde. Einige Sysops sind dann erst mal auf diesem Stand stehen geblieben und wollten abwarten, wie sich die Software weiterentwickelt. Mittlerweile sind viele Neuerungen in die Software eingeflossen und es kommt immer wieder die Frage auf, was sich denn bislang Grundlegendes seit der Version 1.42n geändert hat. Dieses Kapitel soll hierüber Klarheit schaffen und somit helfen von der mittlerweile doch recht alten BCM Version 1.42n endlich wegzukommen!

Bei der Umstellung einer alten Baybox v1.42n auf die aktuelle Version der OpenBCM ist prinzipiell wenig zu beachten.

Alle Mails und bisherigen Konfigurationseinstellungen bleiben unverändert erhalten!

GANZ WICHTIG: beim Umstieg von BCM 1.42n auf die aktuelle OpenBCM v1.05 sind unbedingt alle alten *msg/messages.** Dateien im Verzeichnis *msg* zu löschen, da diese die neue Version zum Absturz bringen können!

Aktuelle *msg/messages.** Dateien sind im Filesurf von DB0ZWI (Amateurfunk), im Filesurf von DB0274 (CB-Funk) oder im Internet auf der Webseite http://dnx274.dyndns.org/baybox zu bekommen. In der aktuellen *msg_*.zip* Datei, die man an o. g. Stellen bekommen kann, sind auch aktuelle *msg/help.** Dateien enthalten, die die Onlinehilfe der Mailbox auf den aktuellen Stand bringen.

Nun ein Überblick über die wichtigsten Änderungen:

- **Maileingabe:**
  - Abbruch einer Mail anstatt nur mit CTRL+X nun auch mit /AB oder /ab möglich
  - Beim Absenden von Mails wird nun wie bei der DPBox auch der spätere Verteilweg angezeigt
  - Reply: Es wird nun auch der neue Titel angezeigt
  - Lifetime-Query wenn LTQUERY in *init.bcm* auf 1 oder 2 gesetzt

- **Mailforward:**
  - Dateiforward funktioniert nun problemlos (Befehle FWDIMPORT bzw. FWDEXPORT)
  - Erweiterung File-Forward: wird in *fwd.bcm* als Connectpfad das Schlüsselwort "FILE:" angegeben, werden Mails automatisch im-/exportiert
  - Handling von Forward-Queue-Dateien verbessert
  - Box kann als reine Forward-Box betrieben werden (Befehl: SFONLY)
  - ausgehender TELNET-Forward nun auch unter Windows möglich
  - Ist eine der Forward-Partner der eigenen Box eine DieBox, so ist bei dieser DieBox in der Datei *mbsys\sfwid.box* folgender Eintrag einzufügen, damit AUTOBIN-Mails richtig übertragen werden:

    BayCom-1.1   18 S
    BayCom-1.2   18 S
    BayCom-   19
    OpenBCM-   19
  - Autorouter integriert (Befehle AUTOFWDTIME und AFWDLIST) als Ergänzung zu den statischen *fwd.bcm*-Definitionen
  - in *fwd.bcm* gibt es neue Optionen bzgl. 7+ und Binärdatei-Unterdrückung

- **User-Datenbank erweitert und neue ALTER Optionen für jeden User**:
  - mit ALTER PS kann man die Optionen für den PS Befehl festlegen
  - mit ALTER PACLEN kann jeder User seine Paclen festlegen
  - ALTER AWAY Funktion neu hinzu
  - ALTER ZIP und ALTER QTH neu hinzu
  - ALTER NOTIFICATION Funktion neu hinzu
  - ALTER FBBCHECKMODE für FBB-Kompatible Checklisten hinzu
  - ALTER REJECT kann auch auf Absender gesetzt werden

- **Ausgaben optimiert:**
  - PS-Ausgabe
  - ST F-Ausgabe zählt übertragene Mails, Statusanzeige verbessert
  - PATH-Ausgabe kommt mit langen H-Path zurecht, zeigt auch WPROT-Routing an

- VERSION zeigt Filesurf-Speicherplatz an, Anzeige aller eincompilierten Features

- **Zahlreiche neue Funktionen in *init.bcm*:**
  (vgl. hierzu auch das Kapitel "23.4. Initialisierungsdatei *init.bcm*")

- callformat	zum Umstellen von CB/Amateurfunkversion
- mailbeacon	Konfiguration der Mailbake
- nopopsmtp	POP3-Mailzugriff ohne vorherigen SMTP-Zugriff
- httpguestfirst	Bei HTTP-Zugriff zunächst immer GAST-Status
- httpaccount	User können selber ein HTTP-Account anlegen
- httpttypw	TTYPW Passwort auch für das AMPRNET verwenden
- defreadlock	Default-Einstellung für ALTER READLOCK
- deffbbcheckread	Default-Einstellung für ALTER FBBCHECKREAD
- ltquery	Lifetime-Abfrage aktivieren/deaktivieren
- unsecurettypw	Konfiguriert den Zugriff auf ALTER TTYPW
- maillistsender	Konfiguriert das Absendercall vom Mailserver
- summertime	Konfiguriert Sommer-/Winterzeit (Windows)
- altboardinfo	Board-Info aus *boardinf.bcm* mit/ohne Return
- timeoutwarning	Timeout-Warnung vor Timeout-Disconnect
- asklogin	Abfrage von Userdaten bei neuen Usern
- tellmode	Konfiguriert/Aktiviert den Tell-Server
- usrtimeout	Legt ein User-Timeout fest
- sfonly	Aktiviert/Deaktiviert reine S&F-Mailbox
- smoothheader	Kompletten oder verkürzten Mailheader ausgeben
- nounknownroute	Mails mit unbekannter Route ablehnen
- addlinuxsystemuser	Support für Linux-User-Password

- **Didadit-Unterstützung:**
  (Didadit-fähige Terminalprogramme sind LinKT, Paxon, WPP und Winstop)
  - Didadit-Transfer-Protokoll Spezifikation v0.91 von DG4IAD/DH3MB implementiert
  - Didadit-Transfer-Protokoll wird nun im Filesurf unterstützt (Up/Download)

- **CB-Funk**:
  - CB-BCMNET Loginkonzept und Erweiterungen implementiert
  - es werden mehr CB-Rufzeichenmuster erkannt

- **TCPIP-Funktionen:**
  - NNTP-Server erweitert
  - neuer FTP-Server auf Port 8021 bietet Filesurf-Zugriff per TCP/IP
  - SMTP File-Attachment werden akzeptiert und in AUTOBIN umgewandelt
  - HTTP-Server-Oberfläche verbessert (u.a. Befehlszeile, /bcm/http/qsl.wav, logo.gif, automatische Passworterstellung für Internet-Boxen...)

- **PW & Hold/Reject:**
  Seit dem Jahr 2000 ist die OpenBCM mit einem erweitertem PW & HOLD/REJECT-System von DF3VI ausgestattet, doch leider wird dies bislang so gut wie gar nicht angewendet. PW & HOLD/REJECT ist ein zweiteiliges System, mit dem sich die Sicherheit der Benutzer gegen Rufzeichenmissbrauch bedeutend erhöhen lässt:
  - Schutz für Benutzer, die noch kein Passwort benutzen
  - Hold-Mechanismus für Nachrichten aus ungeschützter Quelle

  <u>a) Passwortverfahren</u>
  Das geänderte Passwortverfahren wird mit dem Parameter "USERPW 2" aktiviert.
  Damit gelten automatisch eine Reihe Schutzmassnahmen für Benutzer ohne Passwort, was den Zugriff auf Nachrichten betrifft:
   - kein unauthorisiertes Löschen
   - kein Schreiben von (gefälschten) Nachrichten mit Forward

- kein Lesen von fremden Usermails
- kein Verstellen von wichtigen Parametern (wie MyBBS)
- Aufruf von Runutils nur mit User-Passwort (Parameter "-P" in Datei *runutil.bcm* dient dazu, dass Programme nur von Benutzern mit User-Passwort aufgerufen werden können)

In der Datei *msg\pwonly.<sprache>* kann man einen Hinweistext für Benutzer ohne Passwort hinterlegen, der beim Login gesendet wird. Hier sollte man insbesondere darauf hinweisen, wie man ein Passwort bekommt.
Im Gegensatz zum bisherigen Verhalten, wo unter Rufzeichenmissbrauch jeder jedem ein Passwort setzen (und somit aussperren) kann, wird das Passwort nun vom Sysop eingerichtet. Mit den Befehlen PWGEN und SETPW lässt sich eine automatische Passwort-Vergabe durch den Sysop einrichten: Dazu wird mittels PWGEN eine Datei *userpw.bcm* mit 81 Passwörtern vorbereitet, die dann mit "setpw <user> <nr>" Usern zugeordnet werden können. Diese Datei *userpw.bcm* sollte man auch zu Hause haben, um das zugeteilte Passwort dem User mitteilen zu können (natürlich nicht über Funk auslesen!).
Mit dem Parameter "USERPW 3" wird das User-Passwort nicht mehr beim Login, sondern erst später abgefragt - analog zur DP-Box. So wird das User-Passwort nur noch zum Schreiben und Löschen von Mails sowie zum Verstellen von Einstellungen benötigt. Vorteil: ein vergessenes Passwort sperrt einen nicht gleich aus.

b) HOLD-Mechanismus

Der HOLD/REJECT-Mechanismus wird über die Datei *reject.bcm* gesteuert. Zum bequemen Verwalten der Einstellungen gibt es den REJECTEDITOR. In *reject.bcm* kann man parametrieren, wie mit Nachrichten verfahren wird, die ohne Passwort gesendet werden oder über ungeschützte S&F-Wege hereinkommen. Mindestens ein Eintrag "P .B" (kein Bulletin-Send ohne AX25-PW) sollte vorhanden sein. Alle Optionen, die in *reject.bcm* angewendet werden können, sind im Kapitel "5.10. Reject und Hold (*reject.bcm*)" beschrieben.
Wie lange angehaltene Nachrichten nicht geforwarded werden, wird über den Parameter HOLDTIME in Stunden eingestellt (typisch: 48 Stunden = 2 Tage). Mit DIR HOLD lassen sich angehaltene Nachrichten aufzeigen. DIR HOLD funktioniert genauso wie DIR NEWS oder CHECK abhängig vom LastCheck-Timer. Mit "HOLD -u" bzw. "FORWARD -h" kann der Sysop auf Hold gesetzte Mails wieder in den Forward geben.

- **DOS:**
  - Nutzung eines externen Watchdogprogramms möglich
  - Telefonmodem-Support implementiert

- **Linux:**
  - Befehl "addlinuxsystemuser 1" aktiviert ALTER LINUXPW, wo sich User ein Userlogin im Linuxsystem setzen können
  - Befehl "lddversion" zeigt eincompilierte Libs unter Linux an

- **Diverses:**
  - M_Filter-Support implementiert
  - *trace/swaplog.bcm* protokolliert von anderen Mailboxen geswappte Mails
  - Unterscheidung DISC/KILL in SLOG
  - *speech.bcm* zur Verwaltung der Sprachdateien *msg/messages.**
  - Usertimeout durch USRTIMEOUT in *init.bcm* möglich
  - Autosysop-Funktion über *asysop.bcm* implementiert
  - WPROT Funktionen erweitert
  - Wartezeit beim Logout mit Ausgabe des QTEXT reduziert
  - Desaster bei der automatischen Umstellung zwischen Sommer-/Winterzeit gelöst
  - POCSAG-Server-Benachrichtung implementiert
  - QT-Befehl erweitert

- *crontab.bcm* kann nun auch z.B. "*/5" für fünfminütige Aufrufe verarbeiten
- HOLD Befehl erweitert und korrigiert
- statt *convname.bcm* und *convlife.bcm* wird nur noch *convert.bcm* verwendet, wo eine weit bessere Einsortierung von Mails automatisiert werden kann, u. a. nach dem Titel der Mail. Der genaue Syntax von *convert.bcm* ist im Kapitel "5.11. Automatische Rubrikkonvertierung (convert.bcm)" erklärt. Am Anfang kann man seine *convname.bcm* auch erst mal nur in *convert.bcm* umbenennen, die Funktion der alten Einträge bleibt damit erhalten.
- interner *fwd.bcm* und *reject.bcm* Editor (Befehl FWDEDIT, REJEDIT)
- neuer Befehl EXTRACT zum Extrahieren von Dateianhängen einer Mail auf die Mailboxfestplatte, das externes EXTRACT Tool von DF3VI ist damit nicht mehr nötig
- *reject.bcm* um ODER-Verknüpfung erweitert
- Passwort-Verfahren nach DPBox (Befehl USERPW)
- Erweiterung des Conversmodus um Talk-Befehle
- Tell-Funktion wie bei Diebox/DPBox implementiert
- Ping-Funktion neu hinzu
- neuer Befehl UNREAD
- Boxintern wird alles in UTC-Zeit abgewickelt, für den User in Lokalzeit, dadurch u.a. Änderung von logfile Format, evtl. müssen deshalb ein paar alte Runutils angepasst werden (Statistik, ULOG/FLOG...)
- LOG-Befehl überarbeitet, u. a. ist nun eine statische Zusammenfassung möglich
- Bei interaktiven Runutils (d.h. Runutils die auf Usereingaben warten) muss nun in *runutil.bcm* die Option "-A" gesetzt werden

...und viele, viele weitere kleine Verbesserungen und Korrekturen!

In der Dokumentationsdatei *history.txt* ist die komplette Weiterentwicklung der Baybox BCM von v1.00 bis v1.46 dokumentiert. Seit OpenBCM liegt dem Quellcode die Datei *changes.txt* bei, in der alle Änderungen der OpenBCM ab v1.00 dokumentiert sind. Manchmal ist auch ein Blick in diese Dateien sehr interessant.

Zusammengefasst müssen also folgende Punkte beim Upgrade ausgeführt werden:

- Sprachdateien *msg/messages.** und Onlinehilfedateien *msg/help.** ersetzen
- neue Mailbox-Hauptdatei *bcm.exe*, *bcm32.exe* oder *bcm* ersetzt die alte Version
- ggf. *convname.bcm* in *convert.bcm* umbenennen
- die neue Mailbox-Version starten, dabei wird die Userdatenbank automatisch ins neue Format konvertiert
- ggf. die oben aufgelisteten neuen Funktionen konfigurieren und ausprobieren

23. Spezifikation der System-Dateien

23.1. Zielsetzung

Alle Dateien sind darauf ausgelegt, dass die Mailbox insgesamt den folgenden Anforderungen gerecht wird:

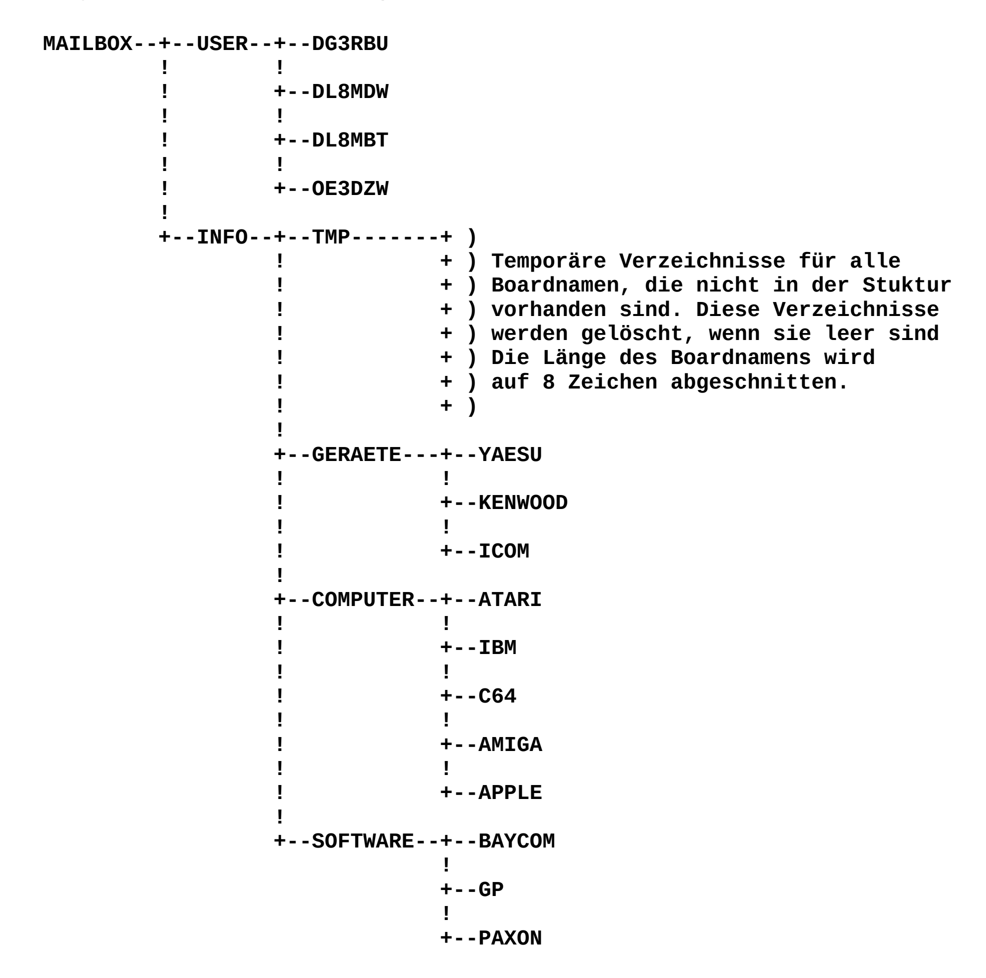
- Robuster, pflegeleichter Betrieb: Durch Betriebsausfälle und Fehlbedienungen dürfen keine Inkonsistenzen in den Daten (insbesondere beim Forwarding-Betrieb) entstehen. Die Dateistruktur muss zu jedem X-beliebigen Zeitpunkt immer restaurierbar bleiben.

- Konstante Transaktionszeit: Unabhängig von der Anzahl der Mails in der Mailbox darf die Verarbeitungsgeschwindigkeit nicht unter ein gegebenes Maß gehen. Gefordert sind:
  - max. 200 ms ununterbrochener Betrieb eines Prozesses
  - max. 1 Sekunde Antwortzeit bei allen einfachen Transaktionen
  - keine Blockierung des Systems durch komplexe Transaktionen (Suchvorgänge etc.) oder Hintergrundoperationen
- Alle Wartungsarbeiten am System müssen ohne Abstellen der Software und aus der Ferne möglich sein.

23.2. Verzeichnis- und Dateistruktur

Für die System-Dateien und -Verzeichnisse der OpenBCM-Mailbox gelten folgende Grundsätze:

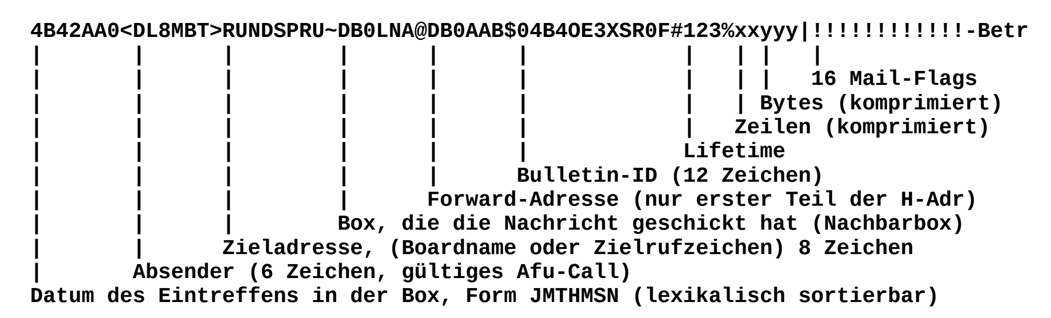
- Alle Verwaltungsdateien haben die Erweiterung *.bcm*
- Die Mails in der Box werden voll auf die Dateistruktur abgebildet. Der dadurch entstehende Overhead wird in Kauf genommen.
- Jeder Benutzer, für den mindestens eine Nachricht vorhanden ist, erhält ein eigenes Verzeichnis.
- Jede Rubrik (Board) erhält ein eigenes Verzeichnis. Der Name dieses Verzeichnisses setzt sich aus maximal 8 Buchstaben zusammen, wobei nur der Dateiname benutzt wird. Die Dateinamen-Extension (nach dem Punkt) bleibt leer.
- Das Anlegen von Unterverzeichnissen darf beliebig geschachtelt werden. Es entsteht ein Verzeichnis-Baum, der voll im Bulletinsystem der Mailbox abgebildet wird. Beispiel:



- Bulletin-Namen, die nicht im (vom Sysop) angelegten Baum vorhanden sind, werden unter dem Verzeichnis TMP temporär angelegt. Sie gehorchen dabei allen Gepflogenheiten der anderen Verzeichnisse, mit dem Unterschied dass sie wieder gelöscht werden, sobald sie leer sind. Diese Automatik greift bei allen Verzeichnissen, die unterhalb von TMP angelegt werden.
- Einschränkung der baumförmigen Struktur: Jeder Bulletin-Name muss eindeutig sein. Bei der Ein- und Ausgabe der Mails verschwindet die baumförmige Struktur und es ist nur noch der Verzeichnis-Name entscheidend. D.h. man braucht beim Wechsel in ein Sub-Board nie den Namen des darüber liegenden Boards mit anzugeben, der Name des Sub-Boards reicht völlig.
- Die Dateinamen der Boxeinträge auf der Dateisystemebene setzen sich aus Datum und Uhrzeit zusammen, wann die Mail in der Box angekommen ist. Beispiel:

```
131N50A
JMTHMSN
      ^ laufende Nummer (0..9, A..Z)
     ^ (Bit0 der Minuten)*15 + Sekunden/4 (0..9, A..Z)
    ^ Minuten/2 (0..9, A..Z)
   ^ Stunden (0..9, A..Z)
  ^ Tag (0..9, A..Z)
 ^ Monat (Hex)
^ Jahr (1=1991, 2=1992, ..., A=2000, B=2001 usw.)
```

  Treffen mehrere Mails innerhalb von 4 Sekunden ein, so wird das letzte Zeichen im Namen hinaufgezählt. Gezählt wird im erweiterten Hexcode, also 0..9, A..Z.
- In jedem Verzeichnis wird eine Datei namens *list.bcm* aufgebaut. Diese Datei ist chronologisch sortiert. In ihr werden alle wesentlichen Daten der Mails in diesem Verzeichnis gehalten und können bei Bedarf ausgelesen werden. Diese Datei ist zwar in ASCII und lesbar, aber sehr formsensibel und deshalb nicht ohne weiteres editierbar. Das Aussehen der Einträge ist folgendes:



```
4B42AA0<DL8MBT>RUNDSPRU~DB0LNA@DB0AAB$04B4OE3XSR0F#123%xxyyy|!!!!!!!!!!!!-Betr
|       |      |        |      |      |            |   | |   |
|       |      |        |      |      |            |   | |   16 Mail-Flags
|       |      |        |      |      |            |   | Bytes (komprimiert)
|       |      |        |      |      |            |   Zeilen (komprimiert)
|       |      |        |      |      |            Lifetime
|       |      |        |      |      Bulletin-ID (12 Zeichen)
|       |      |        |      Forward-Adresse (nur erster Teil der H-Adr)
|       |      |        Box, die die Nachricht geschickt hat (Nachbarbox)
|       |      Zieladresse, (Boardname oder Zielrufzeichen) 8 Zeichen
|       Absender (6 Zeichen, gültiges Afu-Call)
Datum des Eintreffens in der Box, Form JMTHMSN (lexikalisch sortierbar)
```

- Bei User-Mails (und nur bei diesen) wird hinter der Zieladresse ein R (durch ein Blank getrennt) angegeben, wenn die Nachricht vom Inhaber des Zielrufzeichens gelesen worden ist. Da bei Usermails das Ziel max. 6 Zeichen haben kann, ist dafür noch Platz.
- Das ">" vor dem Zielrufzeichen wird gegen ein Blank ersetzt, wenn die Nachricht als gelöscht markiert wird. In diesem Zustand kann die Nachricht wieder restauriert werden (mit dem Befehl UNERASE).
- Die ersten 7 Zeichen in jeder Zeile entsprechen zugleich dem Dateinamen der beschriebenen Mail.
- Die Zahl der Zeilen und die Zahl der Bytes werden in einem komprimierten Format angegeben: Zahlenwerte zur Basis 128, beginnend mit dem !-Zeichen (hex 21 im ASCII-Zeichensatz).
- Die Felder sind exakt einzuhalten. Sind die Feldelemente kürzer als die vorgesehene Länge, so sind Blanks einzufügen. Die Operatoren zu Beginn jedes Feldes werden stets eingefügt, aber nicht ausgewertet. Entscheidend ist allein die Lage der Felder. Zum Ende wird jeder Eintrag mit dem ersten Teil des Betreffs der Nachricht aufgefüllt, so

dass sich eine Gesamtlänge von 128 Bytes ergibt. Der Betreff endet mit der Folge CR/LF.
Achtung: Die Bearbeitung der Datei erfolgt ausschließlich als Binärdatei. Die Datei darf nicht mit CTRL-Z abgeschlossen sein, und es darf keine Konvertierung CR/LF zu CR oder LF erfolgen, um die fixe Strukturlänge von 128 Bytes keinesfalls zu verletzen. Aus diesem Grund ist das Bearbeiten mit einem Editor zu unterlassen!

- Die Kommandos LIST und DIR greifen auf das jeweilige *list.bcm* zu.
- Verzeichnis-Array: Um hinreichende Verarbeitungsgeschwindigkeit zu ermöglichen, wird der gesamte Verzeichnisbaum für die Info-Mails stets im RAM gehalten. Dabei werden die vollständigen Pfadnamen Abgespeichert und die Suche nach einzelnen Verzeichnissen wird zeitoptimiert implementiert. Es wird also bei allen Dateireferenzen nie der gesamte Pfad, sondern immer nur der Name selbst angegeben. Der Pfad kann dann diesem Array entnommen werden. Erforderlichenfalls kann eine Hash-Tabelle nach den Boardnamen hinzugefügt werden, was aber den Nachteil hat, dass nicht nach Namensfragmenten gesucht werden kann. Das Array hat folgende Form:

```
HARDWARE
GERAETE
GERAETE/YAESU
GERAETE/KENWOOD
GERAETE/ICOM
COMPUTER
COMPUTER/IBM
COMPUTER/ATARI
[...]
```

- Die Sortierung erfolgt in der angegebenen Reihenfolge, sprich im Baum von oben nach unten. Dadurch ist sichergestellt, dass bei der Suche nach Fragmenten stets im eigentlichen Verzeichnisnamen eine Übereinstimmung auftreten kann.

23.3. Übersicht über die System-Dateien und -Verzeichnisse

Die Mailbox kann sich in einem beliebigen Verzeichnis auf der Festplatte befinden, üblicherweise wird das Verzeichnis *c:\bcm* (DOS/WinNT) bzw. */bcm* (Linux) verwendet. Die Verwendung eines anderen Verzeichnisses hat keine Auswirkung auf die Funktion der Box. Es ist lediglich unter Linux die Systemvariable "$BCMHOME=/bcm_path" einzustellen, wobei "/bcm_path" jener Pfad ist, in dem sich die Systemdateien der OpenBCM-Mailbox befinden. Es kann dann allerdings passieren, dass manche Runutils nicht mehr korrekt funktionieren.

afwd.bcm

Diese Datei wird vom Autorouter automatisch erzeugt und immer wieder verändert. Hier sind die dem Autorouter bekannten Ziele abgespeichert. Ein manuelles Editieren dieser Datei macht keinen Sinn!

asysop.bcm

In dieser Datei können die Rufzeichen festgelegt werden, die automatisch Sysop-Status in der Mailbox besitzen sobald sie connected sind. Das Format der Datei wird in Kapitel "5.6. Autosysop (*asysop.bcm*)" erklärt.

bcm

Die OpenBCM-Mailbox für Linux. Es können keine Parameter übergeben werden. Es empfiehlt sich außerdem, die OpenBCM-Mailbox mit dem Shell-Script *startbcm* zu starten.

bcm.exe

Die OpenBCM-Mailbox für DOS. Es können folgende Parameter übergeben werden:

- /?: Zeigt mögliche Optionen
- /C: Benutzt bei VGA 25 statt 50 Zeilen (spart 5 kB Speicher bei VGA)
- /M: Benutzt monochrom-Farbattribute
- /N: Es erfolgen keine Zugriffe auf die Videokarte
- /F: Zeigt die mögliche Anzahl offener Dateien; ist dieser Wert kleiner als 40, so ist die Zahl der "FILES" in der Datei *config.sys* zu erhöhen

Für einen ersten Testlauf wird im Unterverzeichnis *msg* mindestens noch die Datei *messages.gb* oder *messages.dl* benötigt.

bcm32.exe

Die OpenBCM-Mailbox für Windows. Nach dem Start öffnet sich ein Fenster mit Trace-Ausgaben. Die Mailbox selbst ist z.B. über Telnet (Standard: TCP-Port 4719) oder HTTP (Standard: TCP-Port 8080) erreichbar.

bct

ein primitives Terminalprogramm für Linux, mit dem man derzeit ausschließlich eine Verbindung zur Box selbst herstellen kann. Während das Programm läuft, darf ggf. die Fenstergröße nicht verändert werden, weil dies nicht abgefangen wird.

beachead.bcm

Falls diese Datei vorhanden ist, so wird deren Inhalt am Anfang jeder Mail-Bake ausgesendet.

beacon.bcm

Enthält die Digipeater-Pfade über welche die Mail-Bake abgestrahlt werden soll. Ist die Bake länger als ein Paket, wird sie in mehreren Paketen ausgesendet. Das Format dieser Datei ist bei DOS und Linux/Windows unterschiedlich!

bidh2.bcm

Diese Datei wird automatisch erzeugt und enthält die 32 Bit-Hash-Tabelle für den Zugriff auf *bids2.bcm* bzw. *bids3.bcm*.

bids2.bcm bzw. ***bids3.bcm*** bei der Linux-Version

Diese Datei wird automatisch erzeugt und enthält die letzten empfangenen und erzeugten Bulletin-IDs. Die Größe dieser Datei darf beliebig anwachsen, bei jedem Reorg wird sie auf MAXBIDS begrenzt.

boardinf.bcm

Diese Datei kann Beschreibungen zu den einzelnen Rubriken enthalten, die bei DIR oder LIST ausgegeben wird. Diese Datei braucht aber nicht vorhanden oder vollständig sein um einen reibungslosen Boxbetrieb zu gewährleisten.

bulletin.bcm

Enthält alle verfügbaren Bulletin-Boards. Dabei beginnen Hauptboards in der ersten Spalte, Subboards sind um ein Leerzeichen eingerückt. Nach dem Boardnamen steht durch mindestens ein Blank getrennt die Lifetime in Tagen. Die Datei wird beim Start der Boxsoftware gelesen und bei einer Änderung erneut auf die Platte geschrieben. Während die Box läuft bringt eine Änderung der Datei nichts, weil die Datei regelmäßig wieder aktualisiert wird.
Achtung: Diese Datei nicht unbekümmert editieren, sondern Änderungen nur über die Befehle LIFETIME, MKBOARD, RMBOARD, MVBOARD vornehmen, da sonst Inkonsistenzen entstehen können.
Wenn die OpenBCM-Mailbox nicht läuft, so ist es ohne weiteres möglich Boards hinzufügen oder eine Lifetime direkt in der Datei zu ändern. Beim Löschen oder Verschieben von Boards muss allerdings sichergestellt werden, dass das Board leer ist, sonst werden dessen Mails unsichtbar

und fallen auch nicht mehr der Lifetime zum Opfer.
Ist die Box leer (vor der ersten Inbetriebnahme), kann die Datei hingegen gefahrlos und nach Herzenslust editiert werden.
In Störungsfällen kann *bulletin.bcm* beschädigt werden. Dann ist es entweder erforderlich, die Datei zu editieren oder ein Backup einzuspielen. Damit dies wirkt, ist folgende Vorgehensweise notwendig:

- disable (kein Logins mehr zulassen)
- mit KILL alle rauswerfen oder SHUTDOWN, damit kein Zugriff mehr erfolgt
- wt bulletin.bcm
- neue Datei einspielen
- SHUTDOWN (ausschalten, da die Datei nur beim Hochstarten geladen wird)

Ist beim Start der Mailbox die Datei *bulletin.bcm* nicht vorhanden, so wird die Rubrikenstruktur aus den vorhandenen Rubriken restauriert. Lifetimes sowie zuvor leere Rubriken gehen dabei verloren. Als Lifetime wird aus Sicherheitsgründen "999" eingestellt.

check.bcm

Diese Datei wird automatisch erzeugt. Hier wird die (sehr große) Liste für den CHECK-Befehl gespeichert. An die Liste werden eintreffende Bulletin-Nachrichten angehängt. Kommt die Liste durcheinander, so kann sie einfach gelöscht werden. Sie wird dann ggf. automatisch neu erzeugt. Dieser Vorgang dauert jedoch eine Weile, das Löschen sollte daher nicht unbekümmert erfolgen. Diese Liste sollte nicht editiert werden!

checknum.bcm

Diese Datei wird automatisch erzeugt und enthält die Board-Nummern für die Checkliste.

convat.bcm

Dient zur Konvertierung von Verteilern. Lokal eingegebene oder im User-S&F empfangene Verteiler werden konvertiert. Beim S&F mit einem offiziellen Forward-Partner wird der Verteiler bei der Auswertung der Zielboxen konvertiert, d.h. alte Verteiler brauchen nicht mehr in *fwd.bcm* stehen. Ein Beispiel ist im Kapitel "5.12. Automatische Verteilerkonvertierung (*convat.bcm*)" angegeben.
Mit dem Befehl CONVAT kann man sich diese Datei ausgeben lassen.

convert.bcm

In dieser Datei können Rubriknamen von ankommenden Mails umbenannt werden. Dadurch lässt sich die Flut der beim Forwarding entstehenden Rubriken eindämmen und es lassen sich Lifetime-Konvertierungen durchführen. Bei der Weitergabe der Nachrichten an die entsprechenden Partner-Forwardboxen werden jedoch wieder die ursprünglichen Rubriknamen verwendet. Der Aufbau und die Funktionsweise der Datei *convert.bcm* ist im Kapitel "5.11. Automatische Rubrikkonvertierung (*convert.bcm*)" näher beschrieben.
Wird der Inhalt der Datei extern verändert, so kann sie mit NEW neu eingelesen werden; bequemer ist die Verwendung der CONVEDIT Funktion.

convlife.bcm und ***convname.bcm***

Diese Dateien wurden in alten Versionen zur Rubrikkonvertierung eingesetzt. Sie werden in den aktuellen Versionen nicht mehr verwendet.

crontab.bcm

Enthält die Zeitpunkte zu denen verschiedene Boxfunktionen aufgerufen werden. Diese Datei sollte während die Box läuft nur mit boxinternen Befehlen (EDIT, RTEXT, WTEXT) bearbeitet werden.

cvusers4.exe

DOS-Programm zum Umwandeln von alten Userdatenbanken (*users.bcm* bzw.

users3.bcm) in das aktuelle Format *users4.bcm*. Die Umwandlung wird normalerweise direkt von der Mailboxsoftware durchgeführt, sofern man denn bei dem gleichen Betriebssystem geblieben ist. Somit wird im Regelfall dieses Tool eigentlich nicht benötigt wird.
Ein Spezialfall ist jedoch der Umstieg von z.B. einer alten DOS-Version, die noch *users.bcm* benutzt, auf die aktuellste Linux-Version, die *users4.bcm* benutzt. Hier kann man dann mit diesem Tool die Datenbank manuell umstellen. Zur Konvertierung ist einfach das Programm im DOS oder Windows-OpenBCM-Verzeichnis zu starten.

fwd.bcm

Hier stehen alle eingestellten Forwardwege. Diese Datei ist komplett vom Sysop anzulegen und wird auch von der Mailbox nicht verändert. Es kann im Betrieb mit dem Befehl NEW neu eingelesen werden. Im Kapitel "9. Forwarding komplett einrichten" wird der Aufbau und die Funktionsweise dieser Datei ausführlich behandelt.

hadr4.bcm

Diese Datei wird automatisch erzeugt und enthält gespeicherte Mailboxadressen. Jeder ankommenden Nachricht werden Daten über durchlaufene Boxen abgenommen, die hier gespeichert werden. Jede Box belegt 1024 Bytes. Diese Datei wird von der Box automatisch erzeugt und kann nicht kann nicht aus vorhandenen Datenbeständen restauriert werden. Sie sollte deshalb regelmäßig gesichert werden!

hadrhash.bcm

Die Datei wird automatisch erzeugt und enthält die 16 Bit-Hash-Tabelle für den Zugriff auf *hadr4.bcm*.

init.bcm

Enthält die Grundeinstellungen der Box (muss also editiert werden!). Während die Box läuft, bringt eine Änderung der Datei gar nichts, weil diese Datei von der Mailbox selbst wieder überschrieben wird. Änderungen müssen bei laufender Box direkt in der Box (als Kommandos) vorgenommen werden, diese werden dann automatisch in *init.bcm* zurück geschrieben.
Siehe dazu Kapitel "23.4. Initialisierungsdatei *init.bcm*".

init.l2

In der Linux- und in der Windows-Version der OpenBCM-Mailbox ist auch die Funktion eines Layer2 enthalten. Dieser wird über diese Datei konfiguriert.
Siehe dazu Kapitel "23.5. Initialisierungsdatei *init.l2*".

issue.bcm

Falls diese Datei vorhanden ist, so wird deren Inhalt vor dem "Login:"-Prompt beim Login per Modem (bei der DOS-Version) bzw. über Telnet (bei der Linux- und Windows-Version) ausgegeben.

macro.bcm

Die Verbindung zwischen Macro-Ereignissen und Skriptnamen wird in dieser Datei festgelegt.
Siehe hierzu auch das Kapitel "30. OpenBCM Skriptsprache".

netpath.bcm

Enthält Connectpfade für das Net-CMD-Interface unter Linux, siehe hierzu auch das Kapitel "19.8. Die Net-CMD-Schnittstelle unter Linux".

passwd.bcm

Enthält die Passwörter für die Sysop-Identifikation mittels BayCom-, MD2- und MD5-Passwort-Verfahren. Für jedes Verfahren ist ein eigenes Passwort anzugeben. Die Passwörter stehen jeweils in einer eigenen Zeile in der angegebenen Reihenfolge.

popstate.bcm
Datei enthält Datenbank zu POP3/SMTP-Zugriffen.

pw
Ein Programm unter Linux, um über Funk einen "root"-Login zu ermöglichen, ohne dass das su-Passwort über Funk gesendet werden muss. Das dabei verwendete Verfahren entspricht exakt dem PW-Verfahren der OpenBCM-Mailbox. Das Programm lebt von der Datei */etc/passwd.pw*, in dem in der ersten Zeile ein 80-stelliger String (wie in *passwd.bcm*) steht, in der zweiten Zeile steht ein Prompt, der vor die 5 ausgegebenen Ziffern geschrieben wird. Das Programm braucht root-Rechte, um zu funktionieren. Das Programm wird nicht benötigt, wenn OpenBCM direkt als root-Prozess ausgeführt wird.

<Rufzeichen>.pwd, z.B. ***db0zdf.pwd***
Diese Datei enthält ggf. ein DieBox-kompatibles Passwort für Forwarding (auch User-S&F). Besser sollte das BayCom-, oder noch besser das MD2/MD5-Passwort-Verfahren verwendet werden.

reject.bcm
Diese Datei enthält die REJECT/HOLD Definitionen, die im Kapitel "5.10. Reject und Hold (*reject.bcm*)" ausführlich erläutert werden.

rhosts.bcm
In dieser Datei können beliebig viele Hostnamen und IP-Adressen angegeben werden (jeweils in einer eigenen Zeile). Wenn das Telnet-Interface der Mailbox von einem Host connected wird, der in dieser Datei angegeben ist, dann kann sich der Benutzer ohne Passwort einloggen und erhält Sysop-Status!
Achtung: Während der DNS-Abfrage steht die Mailbox, hat also der Nameserver eine wackelige Anbindung, so ist es sinnvoll die Anzahl der Einträge in dieser Datei gering zu halten.

rundat.bcm
Wird vor dem Aufruf eines RUN-Utilities erzeugt. Diese enthält Informationen für das aufgerufene Run-Utility.

runutil.bcm
Enthält Informationen über externe Programme, die von Usern aufgerufen werden können, es handelt sich dabei um so genannte Run-Utilities.

speech.bcm
Diese Datei beinhaltet die Sprachdatei-Zuordnungen und wird im Kapitel "5.8. Sprachdefinitionsdatei (*speech.bcm*)" erläutert.

startbcm
Shell-Skript unter Linux zum Starten der Mailbox. Hier werden auch alle notwendigen Umgebungsvariablen vorbereitet, die im Betrieb der Box sichtbar sein sollten. Der Aufruf sollte sinnvollerweise aus */etc/inittab* erfolgen.

userh2.bcm
Diese Datei wird automatisch erzeugt und enthält die 16 Bit-Hash-Tabelle für den Zugriff auf *users4.bcm*.

userpw.bcm
Diese Datei wird durch den Befehl PWGEN automatisch erzeugt und enthält mögliche User-Passwörter, die der Sysop mit SETPW einem User zuordnen kann.

users.bcm und ***users3.bcm***
Diese Dateien wurden in alten Versionen eingesetzt. Sie werden in den

aktuellen Versionen durch *users4.bcm* ersetzt und werden nicht mehr verwendet.

users4.bcm
Diese Datei wird von der Box automatisch generiert und enthält die Daten der bekannten Benutzer, die sie im Laufe der Zeit lernt. Diese Datei kann nicht aus anderen Datenbeständen abgeleitet werden, deshalb sollte von dieser Datei regelmäßig ein Backup erstellt werden!

Verzeichnis ***fwd/***
In diesem Verzeichnis werden für jeden Forward-Partner Dateien angelegt, in denen festgehalten wird, welche Mails zu diesem Partner weitergeleitet werden müssen. Sobald die Dateien nicht mehr benötigt werden, werden sie automatisch gelöscht.

Verzeichnis ***http/***
Hier werden Dateien abgelegt, die über den HTTP-Server abgerufen werden können. Außerdem kann hier *back.jpg*, *logo.gif* bzw. *qsl.wav* ablegen, die das HTTP-Erscheinungsbild verändern können.

http/back.jpg
Diese Datei wird von der Box als Hintergrundbild bei HTTP-Zugriff genutzt, falls vorhanden

http/qsl.wav
Diese Datei wird von der Box als Quittierungston bei jeder HTTP-Aktion ausgegeben, falls vorhanden.

http/logo.gif
Diese Datei wird von der Box als Logo-Bild bei HTTP-Zugriff oben Links angezeigt, falls vorhanden.

Verzeichnis ***msg/***
In diesem Verzeichnis werden alle System-Sprach-Dateien gespeichert. Sie sind ggf. vom Sysop zu editieren und den Gegebenheiten anzupassen. Die Extension <Sprachkenner> bezeichnet den Landeskenner der eingestellten Sprache. Für jede Sprache ist mindestens die Datei *msg/messages.<Sprachkenner>* notwendig.

msg/messages.<Sprachkenner>, z.B. ***msg/messages.gb***
Enthält alle Meldungen der Box in der jeweiligen Sprache.

msg/help.<Sprachkenner>, z.B. ***msg/help.gb***
Enthält die Onlinehilfe. Diese Datei muss existieren, damit HELP Funktioniert.

msg/aktuell.<Sprachkenner>, z.B. ***msg/aktuell.gb***
Enthält den Text, den jeder genau einmal beim Login gezeigt bekommt. D.h. wenn der Text neuer ist als der letzte Login, wird er angezeigt. Mit dem Befehl AKTUELL kann der Text jederzeit angesehen werden.

msg/bnguest.<Sprachkenner>, z.B. ***msg/bnguest.gb***
Enthält den Text, der bei dem Login als Gast im CB-BCMNET Loginkonzept (CB-Funk) ausgegeben wird.

msg/bnpwonly.<Sprachkenner>, z.B. ***msg/bnpwonly.gb***
Enthält den Text, der bei einem Login ohne Passwort im CB-BCMNET Loginkonzept (CB-Funk) ausgegeben wird.

msg/cein.<Sprachkenner>, z.B. ***msg/cein.gb***
Enthält den Text, der beim Login eines Users ausgegeben wird, wenn für diesen User die ASKLOGIN-Abfrage ausgeführt wird (bei "asklogin 1" in *init.bcm*).

msg/cguest.<Sprachkenner>, z.B. ***msg/cguest.gb***
Enthält den Text, der bei dem Login unter GUESTCALL ausgegeben wird.

msg/cnew.<Sprachkenner>, z.B. ***msg/cnew.gb***
Enthält den Text, der beim Login eines Users ausgegeben wird, wenn für diesen User Helplevel 2 eingestellt ist.

msg/cstat4.<Sprachkenner>, z.B. ***msg/cstat4.gb***
Enthält den Text, der bei User-Status 4 ausgegeben wird.

msg/ctext.<Sprachkenner>, z.B. ***msg/ctext.gb***
Enthält den Text, der bei jedem Login eines Users ausgegeben wird.

msg/fshelp.<Sprachkenner>, z.B. ***msg/fshelp.gb***
Enthält den Text, der bei HELP im Filesurf ausgegeben wird.

msg/fsintro.<Sprachkenner>, z.B. ***msg/fsintro.gb***
Enthält den Text, der beim Aufruf des Filesurf-Modus ausgegeben wird.

msg/fsshelp.<Sprachkenner>, z.B. ***msg/fsshelp.gb***
Enthält den Text, der bei HELP im Filesurf ausgegeben wird, falls das eingeloggte Rufzeichen Sysopstatus (erweiterte Befehle) besitzt.

msg/ftpintro.<Sprachkenner>, z.B. ***msg/ftpintro.gb***
Enthält den Text, der bei FTP-Connect als erstes ausgegeben wird.

msg/info.<Sprachkenner>, z.B. ***msg/info.gb***
Dieser Text wird beim Befehl INFO ausgegeben.

msg/nousr.<Sprachkenner>, z.B. ***msg/nousr.gb***
Dieser Text wird bei einem Loginversuch ausgegeben, wenn in der Mailbox der Userzugriff ausgeschaltet wurde.

msg/pwonly.<Sprachkenner>, z.B. ***msg/pwonly.gb***
Dieser Text wird bei einem Login ohne Passwort und einem Wert größer 2 von USERPW in *init.bcm* ausgegeben.

msg/qtext.<Sprachkenner>, z.B. ***msg/qtext.gb***
Enthält den Text, der bei jedem QUIT ausgegeben wird. Dieser Text ist normalerweise sowohl unsinnig als auch unwichtig und sollte deshalb nur bei einem dringenden Bedürfnis eingesetzt werden.

msg/helpidx.<Sprachkenner>, z.B. ***msg/helpidx.gb***
Diese Datei wird automatisch generiert und enthält den Index für die Help-Datei in der jeweiligen Sprache.

Verzeichnis ***user/***
Hier werden alle User-Mails abgelegt. Jeder User erhält ein eigenes Verzeichnis, wenn für ihn Nachrichten vorliegen. Der Verzeichnis-Name kann in *init.bcm* geändert werden.

Verzeichnis ***info/***
Hier werden alle Bulletins abgelegt. Jedes Board erhält ein eigenes Verzeichnis, wenn Nachrichten vorliegen. Der Verzeichnis-Name kann in *init.bcm* geändert werden.

list.bcm
Diese Datei wird automatisch für jedes Board (also in den Verzeichnissen *info/* und *user/*) der Mailbox erzeugt. Sie enthält für jede Mail des Boards Angaben, wie Absender-Call, Titel, etc.

Verzeichnis ***log/***

In diesem Verzeichnis werden tagesweise die Logdateien abgelegt.

log/log<Datum>.bcm, z.B. ***log/log21030.bcm***

Diese Datei enthält das Logbuch für Box-Benutzung und wird von der Mailbox selber angelegt. Pro Tag wird eine Datei erzeugt, die das Datum in der Form JMMTT im Dateinamen hat. Vom Jahr wird jeweils nur die letzte Ziffer verwendet. Beispiel: *log21030.bcm* ist das Logbuch vom 30. Oktober 2002. Der User-Befehl LOG greift auf diese Dateien zu. Die Dateien können problemlos gelöscht werden, wenn sie mit der Zeit zu viel Platz fressen.

Verzeichnis ***macro/***

In diesem Verzeichnis werden die Skripte der Makrosprache abgelegt. Siehe hierzu auch das Kapitel "30. OpenBCM Skriptsprache".

Verzeichnis ***temp/***

In diesem Verzeichnis befinden sich temporär von der Mailbox erzeugte Dateien. Spätestens nach dem Beenden der Mailboxsoftware kann man das Verzeichnis gefahrlos aufräumen.

Verzeichnis ***trace/***

In diesem Verzeichnis befinden sich von der Mailbox erzeugte Dateien, die (Fehler-)Meldungen, Forward-Protokolle und Mitschnitte von User-/Sysop-Eingaben enthalten.

trace/cmdlog.bcm

Hier werden alle vom Sysop eingegebenen Kommandos mit Datum und Uhrzeit gespeichert. Mit dem Sysop-Befehl CLOG kann man sich die letzten zwei kByte dieser Datei ausgeben lassen.
Beispiel:

```
30.09.02 14:21z DH8YMB: par box
30.09.02 14:21z DH8YMB: d b
30.09.02 14:21z DH8YMB: mkb /tmp
30.09.02 14:22z DH8YMB: mkb /baybox
30.09.02 14:22z DH8YMB: reorg
30.09.02 14:22z DH8YMB: purge
30.09.02 14:22z DH8YMB: v
```

trace/eraselog.bcm

Hier werden alle gelöschten Mails protokolliert. Mit dem Sysop-Befehl ERLOG kann man sich die letzten zwei kByte dieser Datei ausgeben lassen.
Beispiel:

```
03.01.03 13:03:08z DBO251: #F DBO274>WP       31DDBO27400Q WP Update
03.01.03 13:35:11z DBX387: #F DBO274>W        31DDBO27400S WP Update
03.01.03 13:58:56z GPN80T: #F DBO274>W        31DDBO27400U WP Update
03.01.03 13:59:36z KS1BBS: #F DGT274>AT5HPK   31DDBO27400T beta20...
03.01.03 14:05:48z DBQ343: #F DBO274>WP       31DDBO27400V WP Update
03.01.03 14:51:15z DBQ946: #F DBO274>W        31DDBO27400W WP Update
03.01.03 15:05:49z DBO812: #F DBO274>W        31DDBO27400Y WP Update
03.01.03 15:06:01z DBX570: #F DBO274>W        31DDBO27400X WP Update
03.01.03 15:35:16z DBO251: #F DBO274>WP       31DDBO27400Z WP Update
```

trace/m_filter.bcm

Hier werden alle M_FILTER Aufrufe mitprotokolliert.

trace/oldumail.bcm

Hier werden alle Usermails mitprotokolliert, bei denen die Lifetime abgelaufen ist, bevor sie vom Empfänger gelesen wurden und bei denen der Absender der Mail mit einer Benachrichtigungsmail über den UNREAD Status der Mail informiert wurde.
Beispiel:

```
27.08.02 03:05:00z D81EUR > SAR0CA @SAR0CA Re: test
26.11.02 03:05:00z DGT274 > DGT882 @DBO274 test
```

trace/pwlog.bcm

Hierbei handelt es sich um die Passwort-Logdatei. Darin werden alle

Passwort-Eingaben mitprotokolliert, egal ob erfolgreich oder erfolglos. Mit dem Befehl PWLOG kann man sich die letzten zwei kByte dieser Datei ausgeben lassen.
Beispiel:

```
30.09.02 12:18z OE3DZW via TCP/IP    tty login failure
30.09.02 12:19z DL8MBT via TCP/IP    tty login failure
```

trace/rejlog.bcm

In dieser Datei werden alle abgelehnten Mails protokolliert. Mit RLOG kann sich die letzten 2 kByte ausgeben lassen.
Beispiel:

```
16.12.02 22:22:25z B DGT274>TEST    @~ $GCCDBO274016 only message to SYSOP without PW
16.12.02 22:24:53z B DGT274>TEST    @~ $GCCDBO274017 only message to SYSOP without PW
16.12.02 22:28:52z B DGT111>TEST    @~ $GCCDBO27401A only message to SYSOP without PW
02.01.03 16:02:30z B PD2RLD>WHY?    @EU~NL3DHG$3352-PD2RLD  board not wanted
```

trace/sfhold.bcm

Hier werden Mails eingetragen, die aufgrund eines Forward-Fehlers nicht weitergeleitet wurden. Mit dem Befehl SFHOLD kann man sich die letzten zwei kByte dieser Datei ausgeben lassen.
Je nach Ursache wird ein Flag gesetzt:

- #R: Reject - Mail wurde vom Forward-Partner mit REJ abgelehnt
- #S: Size - Mail zu Gross (bei gesetzter Option -B in *fwd.bcm*)
- #B: Binary - Forward-Partner unterstützt AutoBIN nicht
- #E: Error - Nachricht ist fehlerhaft (z.B. fehlende BID bei Bulletins)
- #K: kein binär Bulletin Forward (Option -K in *fwd.bcm*)
- #L: Loop - Schleife (nur bei P)
- #M: kein 7plus Bulletin Forward (Option -M in *fwd.bcm*)
- #N: No - eine User-Mail wurde vom Partner abgelehnt, obwohl noch nie angeboten (von Partner welcher nicht REJ unterstützt)
- #H: Hold - eine Mail wird nur aufgrund eines Forwardfehlers auf HOLD gesetzt.

Beispiel:

```
02.01.03 08:54:08z #R DBX387:IT5TNT > SOFT   @ EU $65342_IT5TNT rejected
02.01.03 16:02:38z #R DBX387:RF76WI > TELNET @ EU $5921_RF76WI rejected
02.01.03 16:02:38z #R DBO812:RF76WI > TELNET @ EU $5921_RF76WI rejected
02.01.03 16:02:49z #R NL3DHG:DAA731 > DIVERS @ EU $16425_MG1BOX rejected
02.01.03 16:03:42z #R NL3DHG:DBX353 > Z      @ THEBOX $11DDBX353001 rejected
```

trace/swap.bcm

In dieser Datei werden all die Mails protokolliert, die offensichtlich von einer ANDEREN Mailbox ge"swap"t wurden. Diese Nachrichten bekommen automatisch eine Lifetime von 2 Tagen, während denen der Sysop Nachforschungen anstellen kann und bei Bedarf die Lifetime erhöhen kann. Mit dem Befehl SWAPLOG kann man sich die letzten zwei kByte dieser Datei ausgeben lassen.
Beispiel:

```
23.12.02 10:15:28z AT5HPK>BAYCOM  @DEU~DBQ946$NCCAT5XBR00Z to: BAYBOX
27.12.02 20:51:44z MW18PR>INET    @DEU~GPN80T$NCCLB7BOX00E to: WWW
27.12.02 20:51:44z MW18PR>INET    @DEU~GPN80T$NCCLB7BOX00E @: SAX
27.12.02 20:51:46z MW18PR>PR      @DEU~GPN80T$NCCLB7BOX00D to: PACKET
27.12.02 20:51:46z MW18PR>PR      @DEU~GPN80T$NCCLB7BOX00D @: SAX
01.01.03 02:49:01z AT6HSB>XNET    @DEU~DBO812$11DDBX507005 @: ALLE
01.01.03 02:51:48z AT6HSB>XNET    @DEU~DBO812$11DDBX507006 @: ALLE
```

trace/syslog.bcm

Ablage aller System-Fehlermeldungen. Wenn die Box einen internen Fehler entdeckt, wird er hier vermerkt. Mit dem Befehl SLOG kann man sich die letzten zwei kByte dieser Datei ausgeben lassen.
Beispiel:

```
30.09.02 12:17:52z SYSTEM: #L mbmain: start V1.03 (mem 8876kB, disk 205MB)
30.09.02 12:17:59z SYSTEM: #L cron: crontab.bcm generated
30.09.02 12:18:50z SYSTEM: #L logind: stop (SIGINT)
```

trace/syslog_r.bcm

Bei TRACELEVEL 0 werden hier alle Report-Meldungen abgelegt. Mit dem

Sysop-Befehl SLR kann man sich die letzten zwei kByte dieser Datei ausgeben lassen.
Beispiel:

```
30.09.02 12:17:52z SYSTEM: #R init_tnc: 200 ports
30.09.02 12:17:52z SYSTEM: #R readtree: 0 bulletins
30.09.02 12:17:53z SYSTEM: #R socketd: bcm not found in /etc/services
30.09.02 12:17:53z SYSTEM: #R socketd: initialised to port 4719
```

trace/unknown.bcm

Hier werden unbekannte Forwardadressen vermerkt mit Hinweis auf die Mails, die es betrifft. Mit dem Befehl UNKNOWN kann man sich die letzten zwei kByte dieser Datei ausgeben lassen.
In dieser Datei werden folgende Informationen vermerkt:

- Datum / Uhrzeit
- Login-Rufzeichen (also Boxcall oder Absender)
- Absenderrufzeichen
- Zielrufzeichen oder Boardname
- Forwardadresse (unbekannte Adresse, um die es geht)
- Erster Teil vom Betreff der Nachricht.

trace/t_<Rufzeichen>.bcm, z.B. ***trace/t_hb9w.bcm***

Wenn in *fwd.bcm* für einen Forward-Partner die Option "-T" eingetragen ist, und der Parameter FWDTRACE in der Datei *init.bcm* auf den Wert 2 gesetzt ist, wird das Forwarding mit diesem Partner in dieser Datei genau mitprotokolliert.
Ist in *init.bcm* der Parameter FWDTRACE auf den Wert 1 eingestellt, dann werden alle Forward-Verbindungen mitprotokolliert, unabhängig von der Einstellung in *fwd.bcm*!
Durch Setzten des Parameters FWDTRACE auf 0 wird das Mitprotokollieren von Forward-Verbindungen komplett abgeschaltet.

Die Dateien *users4.bcm* und *hadr4.bcm* werden zwar von der Mailbox erzeugt, können aber nicht aus anderen Daten restauriert werden, falls sie verloren gehen. Sie sollten deshalb regelmäßig gesichert werden!

Die Dateien *bids.bcm*, *bidhash.bcm*, *haddress.bcm, badnames.bcm*, *convname.bcm*, *convlife.bcm*, *users.bcm*, *users3.bcm*, *hadr2.bcm* und *hadr3.bcm* sind Überbleibsel von alten Versionen der OpenBCM/BCM-Mailbox. Sie werden nicht mehr verwendet und können deshalb gefahrlos gelöscht werden.

23.4. Initialisierungsdatei *init.bcm*

Diese Datei enthält die Parameter der Mailbox, die unmittelbar nach dem Start gesetzt werden. Sie können vom Sysop während des Betriebs wie ganz normale Box-Befehle eingegeben werden. Jede Änderung wird sofort in die Datei geschrieben. Eingelesen wird diese jedoch nur beim Start, eine Änderung direkt in der Datei während des Betriebs bringt also nichts.

Alle Parameter der Datei *init.bcm* werden nachfolgend genauer beschrieben:

Abschnitt "Mailbox":

BOXADDRESS	Hierarchische Adresse der Mailbox, z.B.: DB0IRS.#BAY.DEU.EU
BOXHEADER	legt den Text der bei einer "Sent:"-Zeile im Kopf einer Mail der eigenen Box erscheint, z.B.: OpenBCM Oelde JO41BU
CALLFORMAT	Legt Amateurfunk/CB-Version incl. Rufzeichen-Check fest: 0 = Amateurfunkversion, nur Amateurfunkrufzeichen sind erlaubt 1 = CB-Version, nur CB-Funkrufzeichen sind erlaubt 2 = CB-Version, CB- und Amateurfunkrufzeichen sind erlaubt
FSPATH	Pfade, die für User im Filesurf verfügbar sind
GUESTCALL	Legt ein Gastrufzeichen fest, z.B.: GAST
INFOPATH	Pfad, in dem die Bulletin-Mails abgespeichert werden
LOGINCALLFORMAT	Legt Rufzeichen-Check bei Option _LCF in *config.h* fest:

```
                 0 = Amateurfunkrufzeichen sind erlaubt
                 1 = CB-Funkrufzeichen sind erlaubt
                 2 = CB- und Amateurfunkrufzeichen sind erlaubt
MAXBIDS          Anzahl der BIDs/MIDs, die gespeichert werden, z.B. 500000
REMERASE         Erlaubt das Löschen von Mails via Remote Erase
SAVEBROKEN       Konfiguriert den Empfang von 7+ Mails
SYSOPCALL        Call des Sysops - kann sich an der Konsole mit "." einloggen
USERPATH         Pfad, in dem die User-Mails abgespeichert werden
```

Abschnitt "Packet Interface":

```
AX25K_IF         HW-Adresse eines AX.25-Netzwerk-Interfaces (nur Linux)
FWDSSID          SSID, unter dem ausgehende S&F-Connects gemacht werden
MYCALL           AX.25-Calls der Mailbox, z.B. DB0ABC-8 DB0ABC-7
```

Abschnitt "TCP/IP Interface":

```
HTTP_PORT        TCP-Port für HTTP, z.B. 8080
FTP_PORT         TCP-Port für FTP, z.B. 8021
NNTP_PORT        TCP-Port für NNTP, z.B. 8119
POP3_PORT        TCP-Port für POP3, z.B. 8110
SERV_PORT        TCP-Port für das Service Interface, z.B. 8139
SMTP_PORT        TCP-Port für SMTP, z.B. 8025
TELNET_PORT      TCP-Port für TELNET, z.B. 4719
NOPOPSMTP        Legt SMTP-Zugriff fest:
                 0 = SMTP ist nur nach vorherigem POP3 möglich
                 1 = SMTP ist auch ohne vorheriges POP3 möglich
HTTPACCOUNT      Bei "1" ist bei HTTP-Zugriff "create account" aktiv; besser "0"
HTTPGUESTFIRST   Bei "1" ist bei HTTP-Zugriff zunächst GUESTCALL eingeloggt
HTTPTTYPW        Bei "1" wird immer das TTYPW abgefragt, auch für das AMPRNET
UNSECURETTYPW    Konfiguriert den Zugriff auf ALTER TTYPW
```

Abschnitt "Timers":

```
AUTOFWDTIME      Legt die Zeit fest, die vom Autorouter ausgewertet wird
FWDTIMEOUT       Timeout bei S&F-Verbindungen
HOLDTIME         Legt die Haltezeit für Mails mit HOLD-Aktion (reject.bcm) fest
INFOLIFE         Lifetime von Mails in TMP-Board
OLDESTBEACON     Legt fest, wie lange Usermails in der Mailbake angezeigt werden
OLDESTFWD        Maximales Alter von Mails, bis zu dem sie geforwardet werden
USERLIFE         Lifetime der User-Mails
USRTIMEOUT       Legt ein User-Timeout fest
```

Abschnitt "Logging":

```
ERASELOG         Legt fest, in wie weit Löschungen mitprotokolliert werden
FWDTRACE         Legt fest, in wie weit der S&F mitprotokolliert wird
TRACELEVEL       Legt fest, welche Status-Meldungen mitprotokolliert werden
```

Abschnitt "System":

```
DOSINPUT         Definiert die Ausführung eines DOS-Kommandos (nur unter DOS)
DISABLE          Bei "1" werden keine neuen Logins mehr zugelassen
SFONLY           Aktiviert/Deaktiviert reine S&F-Mailbox
TTYMODE          Parameter für die Ansteuerung der seriellen Schnittstelle
USVSENSE         Shutdown der Box durch USV-Warnung (nur unter DOS)
WATCHDOG         Konfiguriert den Watchdog (nur unter DOS)
ADDLINUXSYSTEMUSER
                 Support für Linux-User-Password aktivieren/deaktivieren
```

Abschnitt "User constraints":

```
ALTBOARDINFO     Board-Info aus boardinf.bcm mit/ohne Return ausgeben
ASKLOGIN         Abfrage von Name, QTH, ZIP-Code und MYBBS bei unbekannten Usern
CREATEBOARD      Werden noch nicht vorhandene Boards automatisch angelegt?
LTQUERY          Bei "1" werden Mails nur mit Lifetimeangabe akzeptiert
M_FILTER         Legt ein Mailfilter-Programm fest
MAILBEACON       Bei "1" wird die Mailbake immer gesendet; besser auf "0" setzen
MAXLOGINS        Begrenzt die Anzahl der Logins mit gleichem Call in die Mailbox
```

```
MAXPACLEN        Legt den maximalen User-Paclen-Wert fest (für ALTER PACLEN)
MINPACLEN        Legt den minimalen User-Paclen-Wert fest (für ALTER PACLEN)
NOPURGE          Legt fest, in wie weit Mails beim PURGE gelöscht werden
NOUNKNOWNROUTE   Bei "1" werden Mails mit unbekannter Route abgelehnt
PACLEN           Stellt die globale Paclen ein, bei "0" Paclen-Funktion deaktiv
READLOCK         Gibt ALTER READLOCK für User frei
SMOOTHHEADER     Bei "1" verkürzter Mailheader; besser auf "0" setzen
TIMEOUTWARNING   Bei "1" Timeout-Warnung an den User
USERPW           Legt fest, ob sich User ein Passwort setzen dürfen
USERQUOTA        Anzahl der kBytes, die ein User täglich schreiben/lesen darf
DEFSWAPLF        Legt Lifetime von als ge"swap"ped erkannten Mails fest
```

Abschnitt "Server":

```
MAILLISTSERV     Konfiguriert/Aktiviert den Mailing-Listserver
MAILLISTSENDER   Konfiguriert das Absendercall einer Mailing-Listserver-Mail
OLDUMAIL         Bei "1" Info an den Sender einer Usermail, falls Mail ungelesen
POCSAGLISTSERV   Konfiguriert/Aktiviert den Pocsag-Server
TELLMODE         Konfiguriert/Aktiviert den Tell-Server
```

Abschnitt "User defaults":

```
DEFCHECK         Default-Einstellungen für ALTER CHECK
DEFCMD           Default-Einstellungen für ALTER COMMAND
DEFFBBCHECKMODE  Default-Einstellungen für ALTER FBBCHECKMODE
DEFHELP          Default-Einstellungen für ALTER HELPLEVEL
DEFHOLD          Default-Einstellungen für ALTER FHOLD
DEFIDIR          Default-Einstellungen für ALTER IDIR
DEFILIST         Default-Einstellungen für ALTER ILIST
DEFIREAD         Default-Einstellungen für ALTER IREAD
DEFLF            Default-Einstellungen für ALTER LF
DEFLINES         Default-Einstellungen für ALTER LINES
DEFPROMPT        Default-Einstellungen für ALTER PROMPT
DEFPS            Default-Einstellungen für ALTER PS
DEFREA           Default-Einstellungen für ALTER READLOCK
DEFSTATUS        Default-Einstellungen für ALTER STATUS
DEFUDIR          Default-Einstellungen für ALTER UDIR
DEFULIST         Default-Einstellungen für ALTER ULIST
DEFUREAD         Default-Einstellungen für ALTER UREAD
```

23.5. Initialisierungsdatei *init.l2*

In der Linux- und in der Windows-Version ist ein AX25-Layer2 enthalten. Dieser wird über die Datei *init.l2* konfiguriert. Hier ein Beispiel:

```
;---------------------------------------------------------------------------
; Parameter-Datei für AX.25-Betrieb der BayCom-Mailbox unter Linux und
; Windows
;
; Stand: 25.12.1998
;
; Änderungen für spätere Versionen möglich
;---------------------------------------------------------------------------
dcall    df0ar-8    ; Digicall (normalerweise unnötig)
monitor  1          ; Monitor, siehe BayCom-Terminal 1.60
mselect  0          ; Dito
mcalls   -0         ; Dito
;---------------------------------------------------------------------------
assign   kiss
mode     9600cd     ; Hier die Baudrate angeben ("c" = RMNC-CRC)
;device  COM1       ; Serielle Schnittstelle: COM1 unter Windows
device   /dev/ttyS0 ; Dito: COM1 = /dev/ttyS0 unter Linux
;---------------------------------------------------------------------------
assign   axip
peer     44.130.1.1 ; Hostname oder IP-Adresse für Verbindung über
                    ; AXIP (Adresse des Zielknotens, nicht die eigene
                    ; Adresse!)
                    ; Achtung: Es handelt sich um AXUDP, "normales" AXIP
                    ; ist nicht möglich
```

```
port     4719       ; UDP-Port: Kann beliebig gewählt werden (für
                    ; einen Port unter 1024 sind root-Rechte erforderlich)
;txport  4720       ; Optional: UDP-Port, auf dem gesendet wird, empfangen
                    ; wird immer auf "port"
;-----------------------------------------------------------------------------
; Es sind bis zu 20 Ports konfigurierbar...
;-----------------------------------------------------------------------------
assign  axip
peer    44.130.1.2
port    4721
txport  4722
;-----------------------------------------------------------------------------
```

24. Übersicht über alle Mailbox-Befehle

Im Folgenden findet sich eine Übersicht über alle Befehle der OpenBCM-Mailbox. Die Beschreibung ist jeweils auf eine Zeile begrenzt, deshalb ist sie teilweise unvollständig. Soweit die Parameter noch in der Zeile Platz haben, stehen sie mit dabei, ansonsten wurden sie weggelassen. Parameter in spitzen Klammern müssen angegeben werden, Parameter in eckigen Klammern sind optional. Dabei tauchen folgende Arten auf:

Eine aktuelle Auflistung wird mit dem Befehl CMDLIST bzw. SCMDLIST für Sysopbefehle ausgegeben. Die Verfügbarkeit bestimmter Befehle ist auch von den in der jeweiligen Version eingebauten Optionen abhängig.

<datei>	Für das jeweilige Betriebssystem gültiger Dateiname
<verz>	Für das jeweilige Betriebssystem gültiger Verzeichnisname
<str>	Beliebiger Suchstring, allerdings ohne Wildcards
..	Regulärer Ausdruck, nach dem gesucht wird (RegEx)
<mbox>	Rufzeichen einer Mailbox
<call>	Gültiges Rufzeichen, evtl. mit SSID
<board>	Gültiger Boardname
<addr>	Teil einer hierarchischen Adresse

24.1. Übersicht über alle User-Befehle

#OK#	Wird ignoriert
<GP>:	Wird ignoriert
?	Synonym für HELP
AKTUELL	Gibt den Aktuell-Text aus (z.B. *msg/aktuell.dl*)
ALTER	Zeigt die persönlichen Einstellungen an
-AWAY	Abwesenheitstext aktivieren und definieren
-AWAYENDTIME	Abwesenheitszeitraum beschränken
-CHECK	Voreingestellte Optionen beim CHECK-Befehl
-COMMAND	Befehle, die nach dem Login ausgeführt werden
-DEFAULT	Setzt die ALTER-Einstellungen auf Standard-Werte zurück
-DIRFORMAT	Stellt das Format der DIR/READ/LIST-Ausgaben sein
-ECHO	Schickt die eingegebenen Befehle nach der Abarbeitung zurück
-FBBCHECKMODE	Stellt den FBB-Checkmode ein/aus
-FORWARD	Stellt die Heimat-Mailbox ein
-HELPLEVEL	Stufe der Hilfestellung (z.B. Befehls-Menü beim Prompt)
-IDIR	Voreingestellte Optionen beim DIR-Befehl bei Info-Boards
-ILIST	Voreingestellte Optionen beim LIST-Befehl bei Info-Boards
-IREAD	Voreingestellte Optionen beim READ-Befehl bei Info-Boards
-LF	Anzahl der Leerzeilen zwischen den Befehlen
-LINES	Anzahl der Zeilen bis zum Seitenstop
-LINUXPW	setzt das Linux Systempasswort unter Linux
-LOGINPWTYPE	Passwort-Methode beim S&F (BayCom, MD2, etc.)
-NAME	Teilt der Mailbox den eigenen Namen mit
-NEWCALL	Legt bei Lizenzupdate ein neues Rufzeichen fest
-NOTIFICATION	Legt für eine Mailbenachrichtigung ein Rufzeichen fest

```
 -PROMPT         Ändert das Mailbox-Prompt
 -PW             Ändert den Passwort-String
 -PWLINE         Legt fest, ob vor Passwort-Prompt eine Eingabe erwartet wird
 -QTH            QTH einstellen
 -READLOCK       Schränkt die Lesbarkeit der eigenen User-Mails ein
 -REJECT         Boards, die bei DIR NEWS bzw. CHECK nicht angezeigt werden
 -SPEECH         Sprache, in der die Mailbox-Meldungen erscheinen
 -UDIR           Voreingestellte Optionen beim DIR-Befehl bei User-Boards
 -UFWD           AX.25-Pfad zur eigenen Mailbox (für User-S&F)
 -ULIST          Voreingestellte Optionen beim LIST-Befehl bei User-Boards
 -UREAD          Voreingestellte Optionen beim READ-Befehl bei User-Boards
 -ZIP            ZIP-Code (=Postleitzahl) einstellen
AUTOPATH <call>  Gibt Informationen vom Autorouter aus
BIDLIST [str]    Gibt alle gespeicherten BIDs aus, die "str" enthalten
BIN-RX           Wird ignoriert
BIN-TX           Wird ignoriert
BYE              Synonym für QUIT
CD <board>       Wechselt in das angegebene Board
CHAT <call>      Synonym für TALK
CHECK [str]      Sucht nach einer Mail bzw. gibt die neuesten Mails aus
CHECKCOUNT       Gibt die Anzahl der neuen Mails seit letztem CHECK aus
CMDLIST          Gibt alle User-Befehle aus
COMMENT          Antwort auf eine Mail in die gleiche Rubrik schreiben
CONVAT           Ausgabe der Datei convat.bcm
CONVERS <call>   Synonym für TALK
CONVERT          Ausgabe der Datei convert.bcm
CP               Synonym für TRANSFER
CRONTAB [..]     Ausgabe der Datei crontab.bcm
DATE             Synonym für TIME
DIR              Listet die Mails in einem Board auf
 -AFTER <Datum>  Listet alle Mails auf, die seit <Datum> eingetroffen sind
 -BOARDS [str]   Listet alle vorhandenen Rubriken auf, die "str" enthalten
 -MESSAGES [str] Listet alle Mails auf, die "str" enthalten
 -NEWS           Listet alle Mails seit dem letzten DIR NEWS auf
 -OUTSTAND [mbox]Zeigt die Mails, die noch zum Partner geschickt werden müssen
 -PATH [mbox]    Zeigt die Ziele, die zu den Forward-Partnern geschickt werden
 -SENT <call>    Listet Bulletins auf, die von einem bestimmten Call stammen
 -USERS          Listet alle Calls auf, für die Mails vorliegen
 --ALL           Listet alle bekannten Calls auf
 --LOCAL         Listet alle Calls auf, die schon mal eingeloggt waren
 --MSG           Listet alle User-Mails auf
ERASE            Löscht eine Mail
ERLOG [..]       Ausgabe der Datei trace/eraselog.bcm
EXIT             Synonym für QUIT
F>               Wechselt in den S&F-Modus
FILESERVER       Wechselt in den Filesurf-Modus
FILESURF         Synonym für FILESERVER
FIND <mbox/addr> Synonym für PATH
FINGER <call>    Synonym für USERS
FOLLOWUP         Synonym für COMMENT
FORWARD          Leitet eine oder mehrere Mails an eine andere Mailbox weiter
FS               Synonym für FILESERVER
HEADER           Gibt den Kopf einer Mail aus
HELP             Hilfe (HELP INDEX: Übersicht, HELP ALL: Gesamte Hilfe)
IMPDEL           Wird innerhalb einer IMPORT-Datei verwendet
INFO             Gibt einige Konfigurationsdaten und Datei msg/info.* aus
KOPF             Synonym für HEADER
LIST             Synonym für DIR, jedoch ohne Unterbefehle
LOG              Suche in den LOG-Dateien (Verzeichnis log/*.bcm)
LOGOUT           Synonym für QUIT
LT               Synonym für SETLIFETIME
MAILSERVER       Steuert den integrierten Mailing-Listserver
MAN              Synonym für HELP
```

MD2	Startet die Sysop-Identifizierung mittels MD2-Vefahren
MD5	Startet die Sysop-Identifizierung mittels MD5-Vefahren
MEM	Synonym für STATUS MEMORY
MSG <call>	Synonym für TALK
MYBBS <mbox>	Stellt die eigene Heimat-Mailbox ein (wie ALTER FORWARD)
NAME	Stellt den eigenen Namen ein (wie ALTER NAME)
NEXT	Gibt die nächste Mail aus (bezogen auf das letzte READ)
NH	Synonym für MYBBS
PARAMETER [..]	Ausgabe der Datei *init.bcm*
PATH <mbox/addr>	Info über Weiterleitung einer Mail an eine hier. Adresse
PING	Prüft die Erreichbarkeit einer anderen Box
POCSAGSERVER	Steuert den integrierten Pocsag-Listenserver
PRIV	Sysop-Identifikation mittels DieBox-Passwort (unsicher!)
PS	Listet alle laufenden Mailbox-Prozesse auf
PURGE	Putzt das eigene Board (als Sysop auch ein anderes Board)
PW	Startet die Sysop-Identifizierung mittels BayCom-Vefahren
PWLOG [..]	Ausgabe der Datei *trace/pwlog.bcm*
QT	Setzen des Quit-Datums ohne die Box zu verlassen
QUIT	Beendet die Verbindung zur Mailbox
QUOTA	Ausgabe der noch möglichen Bytes an Transfervolumen
READ	Lesen einer Mail
REJECT [..]	Ausgabe der Datei *reject.bcm*
REPLY	Antwort auf eine Mail an den Absender schreiben
RLOG [..]	Ausgabe der Datei *trace/rejlog.bcm*
RUNUTILS	Listet die verfügbaren Run-Utilities auf
SA	Verschicken einer ACK-Mail
SB	Verschicken eines Bulletins
SEMAPHORES	Synonym für STATUS SEMAPHORES
SEND	Verschicken einer Mail
SETLIFETIME	Ändern der Lebensdauer einer Mail
SF	Anstoßen des User-S&F (als Sysop auch zum Forward-Partner)
SFHOLD [..]	Ausgabe der Datei *trace/sfhold.bcm*
SLEEP	Wartet x Sekunden (nur in IMPORT-Dateien sinnvoll)
SLOG [..]	Ausgabe der Datei *trace/syslog.bcm*
SP	Verschicken einer persönlichen Mail
STATUS	Gibt einige statistische Daten aus
-CPU	Zeigt die CPU-Auslastung an
-FORWARD	Gibt die Anzahl der zum Forward anstehenden Mails aus
-LIMITS	Zeigt Informationen über einige Grenzdaten an
-MEMORY	Gibt eine Liste der verwendeten Speicherblöcke aus
-SEMAPHORES	Ausgabe der gesetzten Semaphoren
SYSOP	Synonym für PW
SWAPLOG	Ausgabe der Datei *trace/swaplog.bcm*
TALK <call>	Schickt einen Text an einen anderen eingeloggten Benutzer
TELL	Einen TELL Befehl zu einer Nachbarbox auslösen
TIME	Gibt das Datum und die (meistens falsche) Uhrzeit aus
TRANSFER	Transferiert eine/mehrere Mails an ein Board/an einen User
UNERASE	Macht eine/mehrere gelöschte Mails wieder sichtbar
UNKNOWN [..]	Ausgabe der Datei *trace/unknown.bcm*
UNREAD	Mail als ungelesen markieren
USERS	Listet die eingeloggten Benutzer auf
VERSION	Gibt Versions-Nummer und andere Informationen aus
WALL	Schickt einen Text an alle eingeloggten Benutzer
WHOAMI	Für alle, die sich ihr Call nicht merken können...
WRITE <call>	Synonym für TALK

24.2. Übersicht über alle Sysop-Befehle

AFWDLIST	Generieren der Autorouter-Datei *afwd.bcm*
APPEND	Hängt eine Zeile an eine Text-Datei an
BATCH	Startet eine *.bat*-Datei und die dazugehörige *.imp*-Datei

BEACON	Sendet die Mail-Bake oder einen beliebigen Text aus
BEGIN <datei>	Gibt die ersten zwei kByte einer Text-Datei aus
CALL <call>	Ändert das Login-Call
CAT	Synonym für RTEXT
CFGFLEX	Sendet Konfigurations-Dateien an einen Flexnet-Knoten
CLOG [..]	Ausgabe der Datei *trace/cmdlog.bcm*
CONNECT	Von der Konsole nach außerhalb connecten
CONVEDIT	Editor für *convert.bcm*
DISABLE	Verhindert weitere User- und S&F-Logins
ENABLE	Hebt den DISABLE-Zustand wieder auf
EXPORT	Leitet die Ausgabe eines Befehls in eine Datei um
EXTRACT	Extrahieren von 7+/BIN einer Mail
FEDIT	Editor für *fwd.bcm* (nur DOS, ohne DF3VI_FWD_EDIT)
FWDEDIT	Editor für *fwd.bcm*
FWDEXPORT	Exportieren einer Datei für Datei-Forward
FWDIMPORT	Importieren einer Datei bei Datei-Forward
GREP	Sucht nach einem regulärem Ausdruck in einer Datei
HOLD	Setzt die angegebenen Nachrichten auf HOLD/setzt HOLD zurück
IMPORT	Führt die Zeilen einer Text-Datei als Befehle aus
KILL	Löscht eine Task (Achtung, evtl. unkontrollierbar)
LIFETIME	Setzt die Default-Lifetime für ein Board
LOGIN	Führt einen Mailbox-Login unter dem angegebenen Call aus
MACRO	Führt ein Makro aus (noch nicht fertig!)
MD2SUM <datei>	Berechnet eine MD2 Prüfsumme von <datei>
MD5SUM <datei>	Berechnet eine MD5 Prüfsumme von <datei>
MKBOARD	Richtet ein neues Board ein
MONITOR	Gibt Ein- und Ausgaben einer bestimmten Task aus
MVBOARD	Verschiebt ein Subboard unter ein anderes Hauptboard
NEW	Liest alle Konfigurations-Dateien neu ein
NOTE	Schreibt eine Nachricht ins Syslog
OCP	Führt den internen COPY-Befehl aus
OMD	Führt den internen MKDIR-Befehl aus
OMV	Führt den internen MOVE-Befehl aus
ORM	Führt den internen RM-Befehl aus
OSHELL	Führt einen Befehl auf Shell-Ebene aus
POSTFWD	Trägt alle User-Mails in die Forwardschlange ein
PWGEN <datei>	Erzeugt eine DieBox-Passwort-Datei
RBIN <datei>	Synonym für RPRG
RDIDADIT <datei>	Gibt eine Datei im DIDADIT-Protokoll aus
REDIT	Editor für *reject.bcm* (nur DOS, falls ohne DF3VI_REJ_EDIT)
REJECTEDIT	Editor für *reject.bcm*
REORG	Führt eine Reorganisation der Mailbox-Daten aus
RMBOARD	Löscht ein Board
RPRG	Gibt eine Datei im AutoBIN-Protokoll aus
RTEXT	Gibt eine Text-Datei aus
RYAPP	Gibt eine Datei im YAPP-Protokoll aus
SCMDLIST	Gibt alle Sysop-Befehle aus
SETPW <call>	Ein mit PWG erzeugtes PW kann einem <call> vergeben werden
SETUSER	Ändert die ALTER-Einstellungen eines Benutzers
SHUTDOWN	Beendet die Mailbox
SLR [..]	Ausgabe der Datei *trace/syslog_r.bcm*
TAIL	Gibt die letzten zwei kB einer Text-Datei aus
TEST	Wird des Öfteren zum Testen neuer Funktionen benutzt
TGREP [..]	Ausgabe der Datei *trace/t_<Mailbox-Call>.bcm*
TTYINIT	Initialisieren der mit TTYMODE definierten Schnittstelle
TNC	Ändert eine Einstellung im AX.25-Interface
UIMPORT	Importiert DieBox-Userdaten (*user3.dat* und *user3.idx*)
WBIN	Synonym für WPRG
WDIDADIT	Speichert eine im DIDADIT-Protokoll empfangene Datei
WPRG <datei>	Speichert eine im AutoBIN-Protokoll empfangene Datei
WTEXT <datei>	Speichert eine Text-Datei
WYAPP	Speichert eine im YAPP-Protokoll empfangene Datei

24.3. Übersicht über alle Sysop-Befehle an der DOS-Konsole

```
CEDIT            Lädt die Datei convert.bcm in den internen Editor
CONNECT <call>   Versucht, eine AX.25-Verbindung zu einem Call aufzubauen
EDIT <datei>     Lädt eine Datei in den internen Editor
FEDIT            Lädt die Datei fwd.bcm in den internen Editor
TWIN             Öffnet ein Fenster mit Trace-Ausgaben
UWIN             Öffnet ein Fenster mit allen eingeloggten Benutzern
W2               Öffnet ein neues Login-Fenster
```

24.4. Übersicht über alle Befehle im Filesurf

```
#OK#             Wird ignoriert
?                Synonym für HELP
BGET <datei>     Gibt eine Datei im AutoBIN-Protokoll aus
BIN <datei>      Synonym für BGET
BPUT <datei>     Speichert eine im AutoBIN-Protokoll empfangene Datei
BYE              Synonym für QUIT
CAT <datei>      Synonym für GET
CD <verz>        Wechselt in ein anderes Verzeichnis
CD..             Wechselt ins übergeordnete Verzeichnis
CHDIR <verz>     Synonym für CD
CMDMODE          Wechselt in den interaktiven Modus
COPY             Kopiert eine Datei
CP               Synonym für COPY
DEL <datei>      Löscht eine Datei
DGET <datei>     Gibt eine Datei im DIDADIT-Protokoll aus
DIR [verz]       Zeigt den Inhalt eines Verzeichnisses an, Wildcards erlaubt
DPUT             Speichert eine im DIDADIT-Protokoll empfangene Datei
EXIT             Synonym für QUIT
GET <datei>      Gibt eine Text-Datei aus
HELP             Gibt die FileSurf-Hilfe aus
LS [verz]        Kurzform des Verzeichnis-Inhalts, mit Parameter "-L" wie DIR
MD [verz]        Legt ein neues Verzeichnis an
MKDIR [verz]     Synonym für MD
MOVE             Verschiebt eine Datei (intern wie COPY + DEL)
MV               Synonym für MOVE
PATH             Zeigt die freigegebenen Verzeichnisse an
PUT <datei>      Speichert eine Text-Datei
QUIT             Beendet den interaktiven Modus
RD <verz>        Löscht ein Verzeichnis
READ <datei>     Synonym für GET
RM <datei>       Synonym für DEL
RMDIR <verz>     Synonym für RD
RPRG <datei>     Synonym für BGET
RTEXT <datei>    Synonym für GET
RYAPP <datei>    Synonym für YGET
TYPE <datei>     Synonym für GET
WPRG <datei>     Synonym für BPUT
WTEXT <datei>    Synonym für PUT
WYAPP <datei>    Synonym für YPUT
YGET <datei>     Gibt eine Datei im YAPP-Protokoll aus
YPUT <datei>     Speichert eine im YAPP-Protokoll empfangene Datei
```

24.5. Übersicht über alle Befehle an den Mailing-Listserver

```
+L               Synonym für NEWLIST
+M               Synonym für ADDMAINTAINER
```

```
+U               Synonym für ADDUSER
-L               Synonym für DELLIST
-M               Synonym für DELMAINTAINER
-U               Synonym für DELUSER
ADDMAINTAINER    Fügt einen Verwalter zu einer Mailingliste hinzu
ADDUSER          Fügt einen User zu einer Mailingliste hinzu
DELLIST          Löscht eine Mailingliste
DELMAINTAINER    Löscht einen Verwalter einer Mailingliste
DELUSER          Löscht einen User einer Mailingliste
DESCRIPTION      Setzt die Beschreibung einer Mailingliste
HELP             Gibt die Hilfe zum Mailing-Listserver aus
INFO             Gibt Informationen zu einer Mailingliste aus
LIST             Listet alle Mailinglisten auf
NEWLIST          Legt eine neue Mailingliste an
OPTIONS          Legt die Optionen für eine Mailingliste fest
RESET            Setzt die Nummer der nächsten Mail einer Mailingliste auf Null
SETNUMBER        Setzt die Nummer der nächsten Mail in einer Mailingliste
SUBSCRIBE        Trägt das eigene Call in eine Mailingliste ein
UNSUBSCRIBE      Trägt das eigene Call aus einer Mailingliste aus
```

24.6. Übersicht über alle Befehle an den Pocsagserver

```
+L               Synonym für NEWLIST
+U               Synonym für ADDUSER
-L               Synonym für DELLIST
-U               Synonym für DELUSER
NEWLIST          Neue Pocsagserver-Liste anlegen
DELLIST          Ganze Pocsagserver-Liste löschen
DESCRIPTION      Beschreibung des Servers
ADDUSER          User zu einer Pocsagserver-Liste hinzufügen
DELUSER          User aus einer Pocsagserver-Liste löschen
INFO             Gibt Informationen zu einer Pocsagserver-Liste aus
LIST             Listet alle Pocsagserver-Listen auf
SUBSCRIBE        Trägt das eigene Call in eine Pocsagserver-Liste ein
UNSUBSCRIBE      Trägt das eigene Call aus einer Pocsagserver-Liste aus
```

25. Benutzung der OpenBCM-Mailbox im CB-Funk

Die OpenBCM-Mailbox ist für den CB-Funk freigegeben. Support-Anfragen im Bereich CB-Funk zu Einstellungen, Installation, Bugreports, Programmierung und Sonstigem werden von Jonas, DO1MJJ/DJJ812, bearbeitet. Jonas ist auch per Email unter "djj812@dbo812.de" zu erreichen. Ebenso ist die Internetseite http://www.dbo812.de/ die offizielle Support-Homepage für die CB-Version der OpenBCM-Mailbox.

Um aus der, nach Installation und erstem Start resultierenden Amateurfunkversion der OpenBCM-Mailbox, eine CB-Funk-Version zu generieren, muss der Parameter CALLFORMAT (und evtl. auch LOGINCALLFORMAT) umgestellt werden. Als Parameter wird eine Ziffer verwendet:

    0 = Amateurfunkrufzeichen / Amateurfunkversion der Mailbox
    1 = CB-Rufzeichen / CB-Funk-Version der Mailbox
    2 = Beide Rufzeichengruppen / CB-Funk-Version der Mailbox

Für CB-Funk muss also "CALLFORMAT 1" oder "CALLFORMAT 2" eingetragen werden.

Ist CALLFORMAT auf 1 oder 2 gesetzt, wird auch "(CB)" als Hinweis für die Aktivierung der CB-Version hinter jeder Mailbox-Versionsangabe mit angezeigt.

Wurde CALLFORMAT auf 1 gesetzt, so werden folgende Zeichenmuster als gültige CB-Rufzeichen erkannt: XXX###, XX####, XXXX##, XXXXX#, XX##XX, XXX##X, XXXX#X,

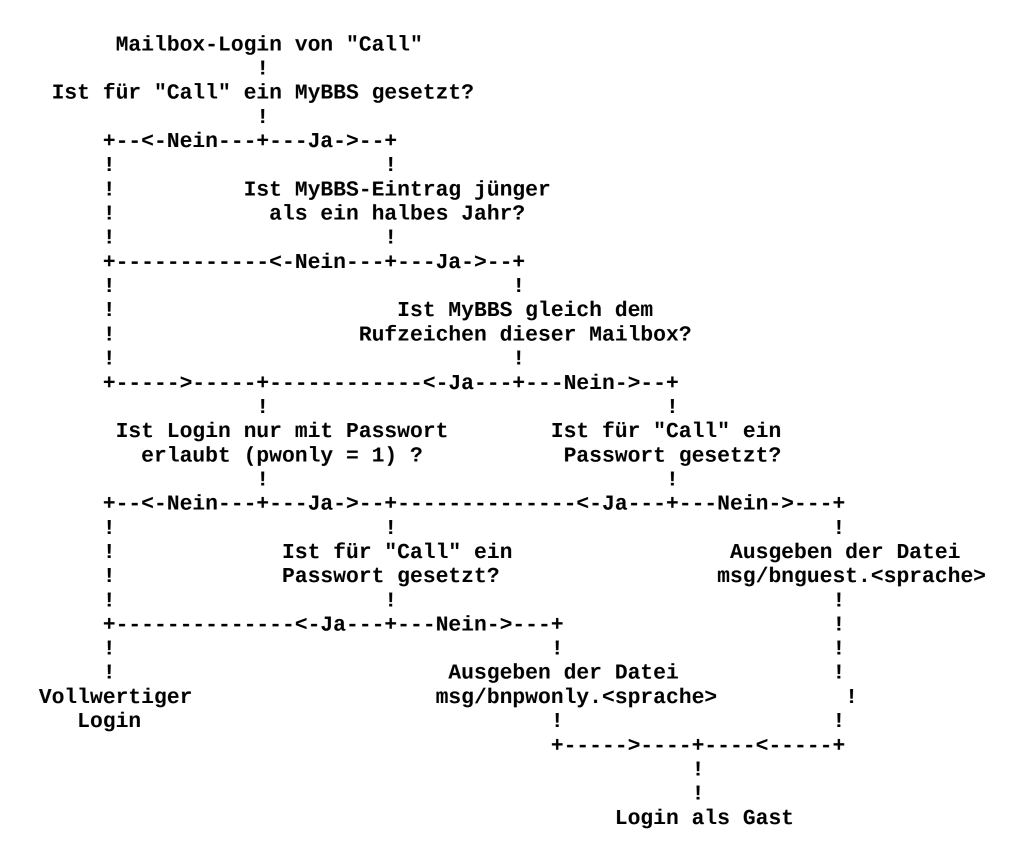
XXX#XX, XX#XXX, XXX#X, XX#XX, XX##X, ##XX## und XXXX#. Ein X steht dabei für einen Buchstaben und ein # für eine Ziffer. Bei CALLFORMAT 2 sind zusätzlich zu diesen Rufzeichen auch noch alle gültigen Amateurfunkrufzeichenmuster freigegeben.

Generell besteht im CB-Funk zwar keine Rufzeichenpflicht, allerdings dürfen keine bereits vergebenen Rufzeichen, seien es angemeldete CB-Rufzeichen, Amateurfunkrufzeichen oder Rufzeichen irgendeines anderen Funkdienstes, verwendet werden! Generell wird die Anmeldung einer Mailbox auf CB empfohlen, um eine gewisse Organisation zu ermöglichen. Mehr hierzu auf der Homepage der RegTP, der Regulierungsbehörde für Telekommunikation und Post, unter http://www.regtp.de.

Ferner ist zu beachten, das Flexnet nicht für den CB-Funk freigegeben wurde, somit ist der Benutzung der OpenBCM unter DOS quasi die Grundlage entzogen!

25.1. Funktionsweise des CB-BCMNET Loginkonzeptes

Im CB-Funk gibt es das Problem, dass die Rufzeichen nicht koordiniert sind und daher mehrfach vorkommen können. Außerdem gibt es Zeitgenossen, welche nichts Besseres zu tun haben als das Netz durch die Benutzung bereits vergebener Rufzeichen durcheinander zu bringen. Das Netz ist damit verwundbar. Aus diesem Grund wurde das CB-BCMNET Loginkonzept mit schaltbarer Sicherheitsstufe (PWONLY in der Datei *init.bcm*) erarbeitet und von Andreas, AS1GBF/DB1RAS, zusammen mit Hannes, AT5HPK, implementiert:



Funktionsprinzip des Loginkonzeptes für das CB-BCMNET:

In eine Mailbox ohne generellen Passwortzugang (PWONLY=0) kann sich nur jeder vollwertig einloggen, wenn

- es sich dabei um ein Rufzeichen handelt, welches in unserem Netz noch nicht bekannt ist, wo also noch kein MyBBS-Eintrag empfangen wurde.
- ein vorhandener MyBBS-Eintrag älter als ein halbes Jahr ist, also davon ausgegangen werden kann, dass dieses Call nicht mehr benutzt wird.
- das MyBBS gleich dieser Mailbox ist, sich der User also in seine Heimatmailbox einloggt.
- auch dann, wenn es sich nicht um die Heimatmailbox des Users handelt, der User aber ein Passwort in dieser Mailbox hat.

Jeder User für den keiner dieser Punkte zutrifft, wird automatisch als Gast eingeloggt. Als Gast hat man vollen Mailboxzugriff, mit folgenden Einschränkungen:

- Alle Mails welche eingespielt werden, bleiben lokal in dieser Mailbox liegen und werden nicht geforwardet, bis der Sysop diese nach Sicht manuell in den Forward stellt.
- Einstellungen wie der Name und die MyBBS werden nicht übernommen, und einige Mailboxbefehle wie erase, transfer,... lassen sich nicht benutzen.
- Alle benutzerdefinierten Einstellungen gelten nur für diesen Login und werden nicht gespeichert.

Bei Mailboxen mit generellem Passwortzugang (PWONLY=1) wird zusätzlich noch geprüft, ob für den User der eines der obigen Punkte erfüllen konnte, ein gültiges Passwort gesetzt ist, ansonsten wird er ebenfalls automatisch als Gast eingeloggt.

Mails welche von einem Gast eingespielt wurden, werden mit

X-Info: Mail was sent from a guest

im Header gekennzeichnet.

Die Dateien *msg/bnguest.** und *msg/bnpwonly.** werden nur bei dem CB-BCMNET Loginkonzept verwendet. Hier sollten Hinweistexte ausgegeben werden, die auf das CB-BCMNET Loginkonzept und seine Restriktionen hinweist.

26. Kompilieren des Quellcodes

Ist der Quellcode erst einmal ausgepackt (unter Windows kann man z.B. WinZip dafür nutzen, unter Linux langt ein "tar xzf sbcm.tgz" am Prompt), sollte man einen Blick in die Datei *config.h* wagen. Hier kann man spezielle Funktionen der Mailbox aktivieren/deaktivieren.

Die Vorauswahl ist unter Linux bzw. Windows meist optimal (es wird fast alles eingebunden), unter DOS muss man hier jedoch meist eine eigene Auswahl der Optionen treffen. Es ist unter DOS nicht möglich, eine Mailbox-Version zu erstellen, die alle Funktionen beinhaltet, da hierfür der DOS-Hauptspeicher nicht ausreicht. Man muss also abwägen, welche Option man nutzen will, und welche nicht. Zum Deaktivieren einer Option reicht es, ein "//" am Anfang der Zeile einzufügen.

Im Kapitel "26.4. Die Optionen in *config.h*" werden alle Optionen der Datei *config.h* erläutert.

26.1. Unter DOS

Unter DOS wird der "Borland C++"-Compiler benötigt. Nach Start des Compilers ist die Projektdatei *bcm.prj* zu öffnen. Kommt beim Kompilieren die Meldung "DLINK exceeds > 64k" müssen weitere Optionen in der Datei *config.h* deaktiviert werden.

26.2. Unter Linux

Unter Linux muss der GCC/GPP GNU Compiler und seine benötigten Komponenten installiert sein. Dieser findet sich bei allen gängigen Linux-Distributionen.

Im Regelfall reicht dann ein "make" im Source-Verzeichnis, um die Linux-Version der OpenBCM zu kompilieren. Bei "make install" wird dann die erstellte Programmdatei in das */bcm* Verzeichnis kopiert. Existiert dort schon eine Datei */bcm/bcm*, wird von dieser Datei eine Backupdatei */bcm/bcm.old* erstellt.
Ändert man Optionen in *config.h*, um diese Auszuprobieren ist es evtl. sinnvoll vor einem neuerlichen "make" ein "make clean" zum Aufräumen des Source-Verzeichnisses zu machen.

Falls beim Kompilieren Warnings (Warnungen) angezeigt werden, kann man diese ignorieren. Treten hingegen Errors (Fehler) auf, so ist meist eine Komponente des GNU Compilers nicht (richtig) installiert. Es ist allerdings auch nicht auszuschließen, dass es sich dann um einen Fehler im Quellcode handelt. Dieser ist aber selten und auf Beta-Versionen beschränkt! ;-)

Es gibt unter Linux noch 3 Spezialfälle:

- bei Verwendung eines alten "SUSE 6.1" Systems sollte man "make -f makefile.suse61" anstelle von "make" aufrufen
- um eine Mailbox ohne Layer2-Support zu erstellen, sollte man "make -f makefile.nol2" anstelle von "make" aufrufen
- um die Mailbox mit dem WX-Modul zu kompilieren (siehe Kapitel "27. WX-Modul in der Linux-Version"), ist "make -f makefile.wx" anstelle von "make" aufzurufen

26.3. Unter Windows

Unter Windows NT/2000/XP wird die Programmierumgebung "Microsoft Visual C++" mindestens ab Version 5.0 oder höher benötigt. Nach Start der Entwicklungsumgebung ist die Workspace *bcm32.dsw* zu öffnen. Im Compiler ist im Menü "Build/Set active configuration" darauf zu achten, dass man eine "Release"-Version der Mailboxsoftware erstellt. Die "Debug"-Version ist für den Dauerbetrieb nicht ausgelegt und kann zu unschönen Effekten führen!

Hinweis: Seltsamerweise scheint sich die deutsche Compilerversion anders zu verhalten als die internationale (englische) Variante. Mit letzterer gab es bislang keine Probleme, während eine Mailbox, die mit der deutschen Variante kompiliert wurde manchmal im Betrieb zu seltsamen Abstürzen neigt.

26.4. Die Optionen in *config.h*

Generelle Optionen:

ALPHA	Der Hinweis "alpha"-Version wird bei der Versionskennung eingeblendet
RUNUTILS	Runutil-Funktion
MAILSERVER	Mailing-Listserver
FILESURF	Filesurf
DF3VI_POCSAG	PocSag-Message-Server
FEATURE_YAPP	Unterstützung für das YAPP-Übertragungsprotokoll
FEATURE_MDPW	Unterstützung für MD2/MD5 Passwörter
DF3VI_FWD_EDIT	Editorfunktionen für *fwd.bcm* (nur für Sysops)
DF3VI_REJ_EDIT	Editorfunktionen für *reject.bcm* (nur für Sysops)
DF3VI_CONV_EDIT	Editorfunktionen für *convert.bcm* (nur für Sysops)
DF3VI_EXTRACT	Extrahieren von 7+/AutoBIN-Teilen aus Mails mit dem Befehl EXTRACT (nur für Sysops)
USERLT	Die Lifetime des Absenders einer Mail wird angezeigt
_AUTOFWD	Automatischer Forward-Router
_GUEST	Gastrufzeichen "GUESTCALL" für TCP/IP und TTY Zugriff
_FILEFWD	Dateiforward-Unterstützung, z.B. für Email-Forward
_LCF	Unterschiedliches Callformat für Userlogin und Forward
DK2UI_TO_FIELD	Ursprüngliche Zielrubrik wird bei LIST/CHECK angezeigt
DF3VI_REJ_LT	Reduzierung der Lifetime bei rejecteten Usermails
DIEBOX_UIMPORT	Importiert Userdaten aus DieBox Datei *user3.dat* und *user3.idx*, wird normalerweise nicht benötigt
_USERS4CONVERT	Modul zur automatischen Konvertierung der Userdatenbank vom alten Format *users.bcm/users3.bcm* ins neue Format users4.bcm
FBBCHECKREAD	Gibt bei CHECK eine Liste mit Mailnummern wie in FBB aus, der READ-Befehl versteht dann auch ein "R <Nr>"
DEBUG_DIDADIT	Füllt Log-Dateien mit diversen Debughinweisen zu DIDADIT, normalerweise nicht sinnvoll
OLDTIMEFMT	Verwendet Zeitformat in Log-Dateien im Format der BCM v1.42, normalerweise nicht mehr sinnvoll

Optionen für Linux und Windows:

NNTP	NNTP-Server
MACRO	Makro-Sprache
SERVIF	Simples Telnet Service-Interface
FEATURE_DIDADIT	Unterstützung für das DIDADIT-Übertragungsprotokoll
_TELNETFWD	Telnet-Forward

Optionen nur für Linux:

FEATURE_SERIAL	Unterstützung für TTY-Login
RADIOIF	Radio Interface (Net-CMD TNT/Wampes)
_AX25K	Unterstützung des Kernel AX.25

Optionen nur für DOS:

SMALL_DOS	Aktiviert eine kleine DOS-Auswahl am Ende von *config.h*
_TNC	Unterstützung für TNC/Flexnet
_TTY	Telefon-Forward
_AUTOTRCWIN	(Auto)Trace-Fenster
FEATURE_EDITOR	Interner Editor
FEATURE_COLOR	Farbauswahl mit ALT-C
FEATURE_MOUSE	Mausunterstützung
BIOS_Y2K	Umgeht Jahr2000-Probleme von alten BIOS-Versionen
OLD_SHELL	Aktiviert die alte DOS-Shell (nicht sinnvoll)
_PREPOSTEXEC	Externer DOS Watchdog (Befehle DOSPREEXEC, DOSPOSTEXEC)

Beta-Optionen:
Werden hier nicht näher erläutert, da sie sich häufig ändern können!

26.5. Rückblick: ein gemeinsamer Quellcode

Die hohe Kunst der portablen Programmierung besteht nicht darin, möglichst verschiedene Versionen für verschiedene Betriebssysteme zu erzeugen, sondern möglichst eine einheitliche Software zu pflegen, die dann auf allen Plattformen übersetzbar und lauffähig ist.

Wird dieses Ziel gut verwirklicht, dann ist auch die Portierung auf weitere Plattformen ohne größere Schwierigkeiten möglich.

So erklärt sich auch, warum es stets nur eine Version gibt und zu einer bestimmten Versionsnummer auch eine bestimmte Funktionalität gehört, ohne dass für die jeweils andere Plattform zusätzliche Arbeit nötig ist.

Trotz dieser guten Vorsätze waren natürlich durchaus Änderungen erforderlich, um den Code portabel zu machen. Folgende Dinge müssen dabei beachtet werden:

- DOS-Dateinamen dürfen groß- oder kleingeschrieben werden. Bei Linux ist die Schreibweise relevant und sollte generell klein sein.
- Integer-Variablen haben unter DOS 16 Bit Darstellungsbreite, bei allen anderen Betriebssystemen sind 32 Bit üblich. Im Einzelfall dürfen keine Annahmen über die Darstellungsbreite getroffen werden. Notfalls müssen Typen definiert werden, die in systemabhängigen Headerdateien definiert werden.
- Systemaufrufe verhalten sich teilweise anders. Sie müssen deshalb in betriebsystemabhängige Module gekapselt werden, um in Richtung zur Anwendung identisch zu sein.
- Hardwarezugriffe müssen vermieden werden. Wenn unbedingt erforderlich, ist eine Virtualisierung sinnvoll, so dass die Anwendung nichts davon sieht.
- Die Programmiersprache muss einheitlich sein. Assemblerteile sind möglichst zu vermeiden, weil diese gewiss nicht portabel sind.

So ergibt sich eine kleine Zahl von Modulen, in denen Abhängigkeiten vom Betriebssystem enthalten sind. Der gesamte Rest (>95% der Gesamtmenge) enthält weder Abhängigkeiten vom Betriebssystem noch von der Hardware.

26.5.1. Gleicher Quellcode für alle Betriebssysteme

Für den Benutzer ergeben sich kaum Unterschiede zwischen den Versionen. Eine wesentliche Zielsetzung bei der Entwicklung der OpenBCM-Mailbox war schon in der DOS-Version, dass für den Benutzer stets eine gewohnte Umgebung mit vorhersehbarem Verhalten existiert. Waren es bei der DOS-Version andere Mailboxsysteme, die als Vorbild dienten, so galt es bei der Linux- und Windows-

Portierung, möglichst viele der bisherigen Bedienungselemente exakt beizubehalten und nicht einzuschränken oder zu verändern.

Diese Zielvorgabe ist auch weitestgehend gelungen, so dass auf der Benutzerseite tatsächlich so gut wie keine Änderungen erkennbar sind. Dies gilt allerdings nur für den tatsächlichen "Benutzer" über Funk. Sysops haben mit sehr wesentlichen Unterschieden, die im Kern des jeweiligen Betriebssystems zu suchen sind, zu rechnen.

26.5.2. Manche Nüsse waren hart zu knacken

Natürlich gibt es einige Programmteile, die sich partout nicht so anpassen lassen, dass sie für alle Plattformen gemeinsam nutzbar sind. Hier mussten echte "Sonderlocken" entwickelt werden, die auch bei zukünftigen Portierungen wieder neu gemacht werden müssen.

26.5.2.1. Ausstieg ins Betriebssystem (Sysop-Shell und Run-Utilities)

Bei DOS ist ein Shell-Ausstieg durch Nachladen von *command.com* möglich. Während eines solchen Ausstiegs steht das aufrufende Programm still. Um einen Ausstieg über Funk möglich zu machen, müssen vorher die Ein- und Ausgaben über die BIOS-Schnittstellen umgelenkt werden. Dies ist kritisch und erfordert sehr systemnahe Eingriffe. Unter Linux erfolgt ein Ausstieg ganz anders und vor allem sehr viel eleganter. Wichtigster Vorteil ist, dass während eines shell-Ausstiegs die Box ohne Behinderung weiterläuft. Die Kommunikation zwischen dem Boxprogramm und der shell (bzw. des ansonsten ausgeführten Programms) erfolgt über ein Pseudo-TTY. Nach Beendigung des aufgerufenen Programms müssen alle davon erzeugten Prozesse wieder sauber aufgeräumt werden.

26.5.2.2. Kontextwechsel im Multitasking-Scheduler

DOS kann bekanntlich nur ein Programm gleichzeitig abarbeiten (von Interrupts abgesehen). Damit die Box viele Benutzer gleichzeitig bedienen kann wurde für die DOS-Version ein Multitasking-Kern entwickelt, der die Anwendung selbst von dieser Aufgabe befreit. Jeder eingelegte Benutzer erhält eine eigene Umgebung und muss sich zum eigenen Ablauf nicht um die Abläufe anderer Benutzer kümmern.

Dieses Subsystem wäre in dieser Art unter anderen Betriebssystemen nicht erforderlich, wenn sog. Threads, also verschiedene Tasks innerhalb einer Anwendung vom Betriebssystem zur Verfügung gestellt werden. Unter Linux wäre dies im Prinzip möglich. Um das Programm so gleich wie möglich zu halten, werden jedoch auch hier die Mechanismen der DOS-Version benutzt. Dies bedingt, neben vielen portabel realisierbaren Mechanismen, die Umschaltung des Stack-Kontextes. Zu diesem Zweck ist genau ein Assemblerbefehl notwendig, der ein Verbiegen des Stackpointers möglich macht. Dieser Befehl ist sowohl von der Hardware, als auch vom Compiler, als auch vom Betriebssystem abhängig und muss deshalb jedes Mal überarbeitet werden.

26.5.2.3. Packet-Radio-Anbindung

Die DOS-Version kommuniziert über einen Software-Interrupt mit Flexnet. Dabei sind nur kompakte Interfaceroutinen in der Box eingebaut, das gesamte AX.25-Handling wird im Interrupt von einem TSR-Programm durchgeführt.

In der Linux-Version ist derzeit der AX.25-Teil mit in der Box enthalten. Dieser kommuniziert nach außen nur über vorhandene Standard-Schnittstellen. Konkret

sind dies CRC-KISS-Mode über serielle Schnittstelle sowie AXIP über UDP für Ethernet-Kommunikation. Außerdem kann die Kommunikation nach außen über ein Linux-Kernel-AX.25-Interface erfolgen.

26.5.2.4. Bildschirm/Tastatur

Bei DOS wird direkt in den Bildschirmspeicher geschrieben. Dies ist notwendig, um eine ansprechende Konsole zur Verfügung zu stellen. Es gibt nur diesen einen Bildschirm, er muss deshalb alle Erfordernisse zur Wartung der Box zur Verfügung stellen. Die Linux-Version wird ganz im Hintergrund betrieben, es findet also keinerlei Bildschirmzugriff statt. Dafür ist es bei Linux vom Betriebssystem möglich, gleichzeitig mehrere virtuelle Bildschirme zu betreiben und von dort über Interprozesskommunikation in die Box einzuloggen.

26.5.2.5.Timer-Interrupt (Erzeugung eines Zeittaktes)

Bei DOS muss zu diesem Zweck ein Interrupt-Vektor auf das eigene Programm gelegt werden.

Bei Linux wird ein zyklisches ALARM-Signal erzeugt, das dann ebenfalls einen Interrupt im Anwendungsprogramm auslöst. Der Zyklus wurde hierbei auf die bei DOS üblichen 55ms festgelegt.

27. WX-Modul in der Linux-Version

In diesem Kapitel wird beschrieben, wie eine WX-Station vom Typ "Peetbros Ultimeter 2000" (siehe http://www.peetbros.com) mit einer Linux-OpenBCM betrieben werden kann. Das WX Modul wurde 1998 von OE3DZW für die Linux-OpenBCM entwickelt. Das Modul kann nicht unter DOS oder Windows verwendet werden!

27.1. Einbinden des WX-Moduls in die Mailbox

Um die WX-Funktion in die Mailbox einzubinden, muss das *makefile.wx* beim Kompilieren verwendet werden. Dies geschieht durch den Aufruf

```
make -f makefile.wx
```

Während des Kompilierens erscheinen evtl. ein paar Kompiler-Warnungen wegen *mbwx.cpp*, diese können aber ignoriert werden.

27.2. Anbindung des WX-Moduls

Die WX Station sollte über einen seriellen Port (z.B. COM1 /dev/ttyS0) mit 2400 Baud an den PC angebunden sein.

Das WX-Modul benötigt ein zusätzliches Unterverzeichnis, z.B. */bcm/wxdata*. Um dies zu erstellen, sollte man eingeben:

```
mkdir /bcm/wxdata
chown bcm:bcm /bcm/wxdata
```

27.3. Konfiguration des WX-Moduls

Nach dem Start der OpenBCM mit einkompiliertem WX-Modul sollte man sich als Sysop in die Mailbox einloggen. Folgende zusätzliche Parameter können nun für das WX-Modul eingestellt werden:

```
wxtty /dev/ttyS0      # für seriellen Port COM1
wxpath /bcm/wxdata    # für den Pfad zu "/bcm/wxdata"
wxstnname <name>      # max. 30 Zeichen
wxqthaltitude <n>     # <n> in Metern
wxsensoraltitude <n>  # <n> in Metern
```

28. Weitere Spezifikationen

28.1. Aussehen einer Mail

Die Nachrichten selbst werden jeweils in einer eigenen Datei abgelegt. Der Dateiname entspricht den im Kapitel "23. Spezifikation der System-Dateien" erwähnten Gepflogenheiten.

Die ersten vier Zeilen einer Nachricht haben organisatorische Bedeutung. Sie werden beim Auslesen der Nachricht grundsätzlich nicht ausgegeben, sondern ausgewertet.

Im Unterschied zu den rein maschinell erzeugten Dateien *list.bcm* und *check.bcm* können die Nachrichten selbst sehr wohl editiert werden. Auch die Form ist relativ tolerant.

Erste Zeile der Nachricht:

```
HUMOR < OE3DZW   DB0LNA @DL $04B4OE3DZW0E #30 %!#!!2 =!!!|B!!!!!!!!!!!!!!
```

Im Prinzip haben die Felder in dieser Zeile exakt die gleiche Bedeutung wie in *list.bcm*. Allerdings wird hier nicht nach Feldlänge ausgewertet, sondern nach den entsprechenden Operatoren ("<", "@", "$", "%", "#", "=", "|") gesucht. Die Reihenfolge ist absolut unerheblich, ebenso die Feldlänge. Blanks dürfen an jeder Stelle eingefügt werden. Dadurch kann jede Nachricht problemlos per Hand editiert und umgebaut werden, es können sogar Nachrichten händisch erzeugt und gelöscht werden.

Die Anzahl der Zeilen und Bytes (nach dem "%"-Operator) hat rein informativen Charakter, sie wird nicht überprüft und anderweitig verwendet, sondern nur beim Befehl READ ausgegeben. Bei der Anzahl der Bytes wird bei AutoBIN-Mails die Länge des BIN-Anteils, bei Text-Mails nur die Nachricht selbst, also ohne R:-Zeilen angegeben. Dadurch ist die angezeigte Größe der Nachricht nicht von der Länge des Forward-Weges abhängig. Der Operator "=" kennzeichnet den Offset des AutoBIN-Teils der Mail, "=!!!" bedeutet, dass die Mail keinen AutoBin-Teil enthält. (Kodierung: "!" entspricht 0, jede Stelle läuft von 0-127 (7 Bit); 7+7+7=21 Bit entsprechen einem maximalen Zahlenwert von 2097151 ("unsigned int" in C), das ist also der maximale Offset). Mit dem Parameter "|" werden 16 Byte-Flags markiert.

Die 2. Zeile betrifft das Forwarding:

```
DB0WGS  OE1XAB  OE5XEM* .......
```

In dieser Zeile stehen Boxrufzeichen, zu denen die Mail geschickt werden soll bzw. bereits geschickt wurde. Ein * hinter jedem Rufzeichen bedeutet, dass die Mail bereits geschickt wurde.

Wichtig ist, dass die Zeile mit einer hinreichend großen Anzahl von Platzhaltern beschrieben ist, da auf die Zeile im Laufe des Forwardingbetriebes schreibende Zugriffe erfolgen, um die Rufzeichen zu markieren. Als Platzhalter sind entweder Blanks oder Punkte zulässig.

Ein "*" am Anfang der Forwardzeile markiert die Nachricht als gelöscht.

In der 3. Zeile wird vermerkt, wer die Nachricht bereits gelesen hat. Das Format ist ähnlich wie bei den Forwardcalls, es wird allerdings nur ein Blank zwischen den Rufzeichen freigelassen. Als Platzhalter werden auch hier Punkte verwendet. Diese Zeile hat eine Länge von 79 oder 252 Zeichen (je nach Version).

In der 4. Zeile steht der Betreff der Nachricht. Dieser darf im Prinzip beliebig lang sein, muss jedoch in einer Zeile stehen.

In der 5. Zeile steht "From:", gefolgt vom Absendercall und Call der Box, in der die Mail eingegeben wurde:

```
From: <absender-call> @ <boxadr>
```

Stimmt die Absendermailbox <boxadr> nicht mit der MyBBS des Absenders überein, d.h. wird die Mail nicht von der MyBBS aus gesendet, so wird eine zusätzliche Zeile eingefügt:

```
Reply-To: <absender-call> @ <absender-mybbs>
```

Beim Empfang einer Mail wird der Name in der From-Zeile ausgewertet und falls der Name des Absenders im *users.bcm* noch nicht bekannt ist, dort eingesetzt. Das Rufzeichen <absender-call> in dieser Zeile muss mit dem Absenderrufzeichen übereinstimmen, sonst wird der Name ignoriert. Es wird neben "From:" auch "de" (DieBox) ausgewertet.

In der nächsten Zeile steht "To:", gefolgt vom Empfängercall und der vollständigen Empfängeradresse.

Wird die Nachricht im User-S&F empfangen, so wird eine Zeile

```
X-Info: Received in User S&F from OE3DZW at OE3XSR.#OE3.AUT.EU
```

eingefügt. Wird die Mail von einem User eingespielt, der sich an der Einspielbox kein Login-Passwort gesetzt hat, so wird eine Zeile

```
X-Info: No login password bzw.  X-Info: No upload password
```

zusätzlich eingefügt. Die X-Info-Zeilen werden beim Empfang nicht ausgewertet.

Die nächste Zeile ist immer eine Leerzeile.

Hinweis: In früheren Versionen war die Länge des Headers fix, damit war die Leerzeile (und damit der Beginn des Inhalts der Mail) ebenfalls fix in der 8. Zeile.

In der nächsten Zeile beginnt der Inhalt der Nachricht.

Zusammengefasstes Beispiel:

```
HUMOR < OE3DZW ~ DB0LNA @DL $04B4OE3DZW0E #30   %!#!!2 =!!!
DB0WGS OE1XAB OE5XEM ....................................bis Pos. 79
.........................................................bis Pos. 252
Langweiliger Titel
R:941104/0119z @:OE3DZW.#OE3.AUT.EU [BayCom-Mailbox] bcm1.39
From: OE3DZW @ OE3DZW.#OE3.AUT.EU (Dietmar)
To  : HUMOR @ DL
Reply-To: OE3DZW @ OE1XAB.#OE1.AUT.EU
X-Info: No login password

Eine kurze Mail <CR><LF>
```

Eine Mail mit einem AutoBIN-Teil hat ein ähnliches Aussehen:

```
HUMOR < OE3DZW @OE3XSR.#BAY.DEU.EU       $04B4OE3DZW0F #123 %!#!!, =!%}
DB0WGS ..................................................bis Pos. 79
.........................................................bis Pos. 252
Ein kurzweiliger Titel
R:941104/0120z @:OE3DZW.#OE3.AUT.EU [BayCom-Mailbox] BCM1.39
From: OE3DZW @ OE3DZW.#OE3.AUT.EU (Dietmar)
To  : HUMOR @ OE3XSR.#BAY.DEU.EU

Das ist der Text einer binären Mail.<CR><LF>
#BIN#10#|43301<CR><NUL><NUL>...<NUL>
```

Direkt anschließend folgt der binäre Teil der Nachricht

```
<NUL> = 0x00, <CR> = 0x0D, <LF> = 0x0A
```

Beim Empfang von Mails wird die Zeichenfolge "//e " am Zeilenanfang erkannt und der erste "/" gegen "\" ersetzt. Dies soll die Verbreitung von Mails unterbinden, die Terminals durch den "//ECHO"-Befehl dazu veranlassen, beliebige Befehle an die Mailbox zu schicken.

28.2. Mails mit AutoBIN-Teil

In der OpenBCM-Mailbox ist ein zur DieBox kompatibles AutoBIN eingebaut.

Unter binären Mails werden Nachrichten verstanden, welche eine beliebige Zeichen in einer beliebigen Reihenfolge enthalten dürfen. Das Übertragungsprotokoll beruht darauf, dass die Dateilänge am Anfang der Übertragung angegeben wird. Die Datei selbst wird transparent übertragen.

28.2.1. Senden von AutoBIN-Mails

Der Befehl SEND hat die gleiche Syntax wie bisher. Wird jedoch ein "<CR>#BIN#" im Datenstrom erkannt, sucht die OpenBCM-Mailbox nach Zahlen von 0-9, die nach dem zweiten # folgen. Die damit dargestellte Anzahl von Zeichen wird eingelesen und sodann die Datei geschlossen.

Die generelle Zeichensequenz zur Einleitung eines AutoBIN-Transfers lautet:

<CR>#BIN#<länge>#|<CRC>#Dateiname<CR>

Verbindlich ist dabei allerdings nur das "#BIN#" sowie die Längenangabe. Der Dateiname wird nicht ausgewertet, sondern transparent durchgereicht.

Beispiel:

S OE3DZW BINTEST<CR>
Es folgt ein kurzer Binärteil<CR>

(Hier startet der Benutzer die AutoBIN-Übertragung bei seinem Terminalprogramm, der Rest geht automatisch.)

-> Terminal-Programm:
<CR>#BIN#10#|23412#$1234567#C:\BOX\TEXTE\BIN.EXE<CR>

-> Box: #OK#<CR> (nicht im S&F)

-> Terminal-Programm: 123456789<CR> (Das sind genau zehn Zeichen!)

Wichtig: Die "#BIN#"-Sequenz wird komplett gespeichert, die OpenBCM-Mailbox wertet aber nur die in der Sequenz übermittelte Anzahl von Zeichen und die CRC aus. Die Sequenz ist auf eine Zeile beschränkt und muss mit einem <CR> (HEX 0D) abgeschlossen werden. Das nächste Zeichen nach dem <CR> ist das erste Zeichen des Binärteils. Die Anzahl der Zeichen des Binärteils muss unmittelbar nach der "#BIN#"-Sequenz folgen und darf nur die Ziffern '0' bis '9' enthalten. Bei jedem anderen Zeichen wird die Auswertung der Längenangabe abgebrochen.

Die CRC wird nach einem von DC4OX in den Boxen veröffentlichten Verfahren berechnet. Wurde keine CRC mitgesendet, so wird von der OpenBCM-Mailbox diese selbständig berechnet und hinzugefügt. Falls die berechnete mit der übermittelten CRC nicht übereinstimmt, so wird eine Fehlermeldung ausgegeben und die Nachricht nicht abgespeichert.

Das 16 Bit-Prüfpolynom bietet nur einen recht beschränkten Schutz gegen Übertragungsfehler. Es können damit Einzelbitfehler nur bis zu einer Größe von 8 kB sicher erkannt werden. Bei größeren Dateien können Einzelbitfehler unerkannt bleiben, die erreichte Übertragungssicherheit ist jedoch immer noch höher als

ohne jede Prüfsumme, zumal meist nicht Einzelbitfehler auftreten, sondern von Digis oft Teile von Paketen verschluckt werden - womit es zur "Fehlersicherung" ausreicht die Dateilänge zu vergleichen – und diesen Zweck erfüllt das 16 Bit-Polynom.

In den Mail-Dateien wird der Binär-Start in der Headerzeile nach dem Operator "=" im komprimierten Format angegeben. Am Binärstart ungleich "=!!!" (entspricht 0) erkennt man eine AutoBIN-Mail. Sie darf keinesfalls mit einem Editor bearbeitet werden, da sonst evtl. der Inhalt oder der Offset nicht mehr passt. Am Beginn des eigentlichen Binärinhalts wird die #BIN#...-Zeile gespeichert und auf insgesamt 80 Zeichen mit <NUL> (nicht blank) aufgefüllt.

Der Titel im *list.bcm* ist mit dem Zusatz "(BIN)" versehen. Dadurch unterscheidet sich der Titel im *list.bcm* von dem in der Nachricht selbst. Der Zusatz "(BIN)" wird auch nicht geforwardet.

Hier ein für die Berechnung der Prüfsumme Implementierungsbeispiel in 'C':

```
/*
 *   CCITT Polynom x^16 + x^12 + x^5 + 1
 *
 *   hier im Gegensatz zu CCITT, (A)X.25:
 *     - Datenbyte MSB zuerst
 *     - CRC-Start mit 0
 *     - keine Multiplikation Nachricht mit x^16
 *       (CRC wird bei Empfang nicht über CRC-Bytes berechnet)
 *     - kein Invertieren des CRC
 *
 *   wird z.B. benutzt von:  UoSat DCE, SEVEN
 */
unsigned crctab[256];
unsigned crc;
unsigned byte;        /* Datenbyte, high Byte = 0 ! */
/*
 *   entweder beim Programmstart einmal aufrufen,
 *   oder die 512 Byte Tabelle einmal ausrechnen lassen und dann als
 *   initialisierte Tabelle ins Programm übernehmen
 */
void init_crctab(void)
  { unsigned n;
    unsigned m;
    unsigned r;
    unsigned mask;
    static unsigned bitrmdrs[] = { 0x9188,0x48C4,0x2462,0x12*,
                                   0x8108,0x4084,0x2042,0x1021  };
    for (n = 0; n < 256; ++n)
      { for (mask = 0x0080, r = 0, m = 0; m < 8; ++m, mask >>= 1)
          if (n & mask) r = bitrmdrs[m] ^ r;
/*
 *       "if (n & mask) r ^= bitrmdrs[m];"  ist kürzer, aber
 *       manche Compiler übersetzen das falsch
 */
        crctab[n] = r;
      }
  }
/*
 *   byteweiser Algorithmus, C, portabel
 */
void do_crc(void)
  {
    crc = crctab[crc >> 8] ^ (crc << 8 | byte);
  }
```

28.2.2. Lesen von AutoBIN-Mails

Hier ist die meiste Arbeit durch die Terminalprogramme zu erledigen.

Von der OpenBCM-Mailbox aus ist das Auslesen einer Nachricht mit Binärteil kaum anders als das normale Auslesen einer Nachricht. Sie erkennt intern den binären Zustand und unterbindet damit jegliche Unterdrückung nicht erwünschter Zeichen. Das ist alles!

Wichtig für Programmierer von Terminal-Programmen: Die "<CR>#BIN#..."-Sequenz kann auch über Frame-Grenzen hinweg beim Terminalprogramm ankommen kann. Die Terminalprogramme sind verantwortlich für die Erkennung der Sequenz und haben entsprechende Vorbereitungen zu treffen, um den Binärteil auf die Festplatte zu schreiben.

Wichtig ist für den Binärteil nur das <CR>#BIN#<NUMMER><CR>. SP z.B. benutzt noch weitere Elemente, die auch mit ausgesendet werden, sie sind für die eigentliche Übertragung des Binärteils aus der Sicht der Mailbox jedoch ohne Belang.

Wichtig ist weiterhin, dass vor dem Binärteil einer Nachricht noch ein beliebiger anderer Text erscheinen kann. Die Terminalprogramme müssen also auf die #BIN#- Sequenz warten, und dürfen den Status nicht ändern, es sei denn, der Benutzer verlangt es.

Es kann immer nur ein Binärteil in einer Nachricht übermittelt werden.

Den Abschluss des Binärteils bildet die Zeichenfolge <CR>#OK#<Prüfsumme><CR> der von der Box ausgesendet wird (nicht im S&F).

Falls das Terminalprogramm "#OK#", "BIN-RX" oder "BIN-TX" aussendet, so wird dies von der Box ignoriert.

Die Option "-X" beim READ von AutoBIN-Mails bewirkt, dass nur der Textteil der Nachricht ausgegeben wird. Das Lesen und Schreiben einer AutoBIN-Mail kann nicht abgebrochen werden.

28.2.3. Forward von AutoBIN-Mails

Binäre Nachrichten werden genauso wie gewöhnliche Text-Nachrichten entsprechend der Eintragungen in *fwd.bcm* weitergeleitet. Beherrscht die Nachbarbox AutoBIN nicht, so wird eine Nachricht erzeugt, welche angibt, wo die AutoBIN-Mail liegt.

Aktiviert wird AutoBIN beim Forward durch eine der Bedingungen:

- Option "-I" in *fwd.bcm*
- SID-Kennung D

Ist eine der Forward-Partner der eigenen Box eine DieBox, so ist bei dieser DieBox in der Datei *mbsys\sfwid.box* folgender Eintrag zu machen:

```
BayCom-1.1  18 S
BayCom-1.2  18 S
BayCom-     19
OpenBCM-    19
```

Damit ist sichergestellt, dass ab der BCM-Mailbox 1.30 und der OpenBCM auch der Forward von AutoBIN-Mails funktioniert.

Wird beim Forward ein CRC-Fehler erkannt, so wird dieser durch das Prompt "|>" statt ">" gekennzeichnet. Die Mail wird dann bis zu viermal wiederholt.

28.3. Beschreibung Forward (Ablauf)

Das Forward-Protokoll wurde nie definiert. Ausgehend von W0RLI's Mailbox [7] wurden von mehreren Autoren div. Protokollvarianten entwickelt. In der OpenBCM-Mailbox wurden die Erweiterungen von DF3AV, F6FBB und DL8HBS [2] implementiert.

Das in der OpenBCM-Mailbox implementierte Forwardprotokoll ist zumindest zu folgenden Mailboxsystemen kompatibel: W0RLI, DieBox, FBB, DPBox, MSYS, diverse

TNC-Mini-Mailboxen. In [11] befindet sich eine ausführliche Beschreibung des FBB und DieBox-Protokolls.

28.3.1. System-Identifier (SID)

Nach dem Verbindungsaufbau beim Forward tauschen die beiden Mailboxen Zeichen aus, welche als SID (**S**ystem **ID**entifier) bezeichnet werden. Dieser SID dient zum Aushandeln des Forwardprotokolls. Das beim anschließenden Forward verwendete Protokoll wird durch die Menge der von beiden Forward-Partnern unterstützen Features bestimmt.

Minimalkonsens:

- Der System-Identifier ist folgendermaßen strukturiert: "[f1-f2-f3]"
- Die Bindestriche (-) kennzeichnen das Ende des ersten Felds und den Anfang des letzten Felds.
- f1, f2 und f3 dürfen die Zeichen "[" oder "]" nicht enthalten.
- f1 ist die Identifizierung für den Autor der Software.
- f2 enthält softwarespezifische Daten. Es kann enthalten, was sich der Autor wünscht, zum Beispiel die Versionsnummer der Software. Dieses Feld darf Bindestriche (-) enthalten.
- f3 ist das Set von unterstützen Features. Es darf keine Bindestriche enthalten. Es enthält einen String von nicht-numerischen Zeichen, jeweils ein Zeichen für ein von der Box unterstütztem Feature. Jedes Zeichen kann auch von Ziffern gefolgt werden, diese geben die Revisionsnummer des Features an. Das Fehlen der Ziffer(n) ist mit der Revisionsnummer 0 gleichzusetzen.

Die OpenBCM-Mailbox sendet folgendes SID:

```
[OpenBCM-1.05-AB1D1FHMRW$]
           ^Unterstützte Features
       ^Versionsnummer
 ^Name der Software
```

Folgende Features sind bekannt:

- A: Ack-Mails werden gesendet und verarbeitet
- B: Komprimierter Forward nach F6FBB
- B1: Resume-Mode beim Forward nach F6FBB
- D: DieBox-Erweiterungen
- D1: Block-CRCs beim Forward nach F6FBB
- F: Forward mit Fünferwechsel nach F6FBB
- H: Hierarchische Adressierung
- M: MIDs werden unterstützt (BIDs bei persönlichen Mails)
- R: Unterstützung erweiterter Reject-Meldungen
- W: Unterstützung von WPROT nach OE3DZW
- Z: Unterstützung des "CB-BCMNET Loginkonzeptes" (CB-Funk)
- $: BIDs (Bulletin-IDs) werden unterstützt

Das Feature D ist wie folgt definiert:

- Achtstellige Rubriknamen (Empfangsadressen bei Bulletins) sind erlaubt.
- Lifetimes werden im Box-Forward weitergeleitet.
- Forwarding von binären Mails nach dem AutoBIN-Protokoll.
- Forwarding von Erase- und MyBBS-Mails in einem besonderen Format.

Der Buchstabe D wurde von DL8HBS vorgeschlagen, er bedeutet so viel wie "D"L8HBS oder "D"ieBox. Die D-Features werden derzeit von folgenden Mailboxen verwendet:

- BayCom/OpenBCM-Mailbox (DL8MBT, OE3DZW, DG9MHZ, DH3MB, DK2UI, DH8YMB)
- Digi-Point-Mailbox (DL8HBS)
- DieBox (DF3AV, DL1BDY) - jedoch derzeit teilweise ohne "D" im SID

- WinGT-Mailbox (DG8NDL)
- MCuT-Mailbox (DG6VJ, DG4IAD)

Die Existenz des SIDs impliziert, dass das System die Richtung des Forwards ändern kann und OK/NO-Meldungen erzeugt.

Die OpenBCM-Mailbox wertet jedes empfangene SID aus. Bei der OpenBCM-Mailbox wird untersucht, ob das $-Zeichen in der SID enthalten ist. Falls nicht, erfolgt ein Disconnect, weil Systeme ohne BID nicht unterstützt werden.

Im SID anderer Boxen kann es auch folgende Zeichen geben:

- A (F6FBB): ACK-Messages sind bekannt (Forward-Befehl SA)
- B (F6FBB): nur in Verbindung mit F: Huffman-komprimierter S&F
- B1 (F6FBB): wie B, aber mit CRC am Beginn der Nachricht, zusätzlich Resume-Modus
- C (CBBS): Automatische Systemuhreinstellung (obsolet)
- D (DL8HBS): Rubriknamen achtstellig, CRC-16, in Verbindung mit B auch gepackt, CRC verbessert (in Verbindung mit B1)
- D1 (DL8HBS): wie D, zusätzlich Block-CRC-Mode im komprimierten Forward
- F (F6FBB): Fünferwechsel (Batch forward), F6FBB-Syntax verwenden
- H (W0RLI): System unterstützt hierarchische Adressen.
- I (W0RLI): System unterstützt ein "Null-Kommando", eine Zeile welche mit einem Strichpunkt (;) beginnt.
- L (G1NNA): Komprimierung (obsolet)
- M (W0RLI): Unterstützt MIDs (BIDs bei Usermails)
- R (AA4RE): Unterstützt erweiterte Reject-Meldungen
- X (W0RLI): X-Forward
- Y (WA7MBL): Forward mit dem binären YAPP-Protokoll
- $ (WA7MBL): Unterstützt Bulletin IDentifier; das Dollar-Zeichen muss der letzte Buchstabe im Feld f3 sein!

Einige Beispiele von SIDs:

- [RLI-17.5-HIX$]
  w0rli v17.5, unterstützt BID, H-Adr., Null-Kommandos und den X-Forward
- [CBBS-5.1-$]
  ag3f Version der rli/gyq cbbs.
- [MSYS-1.16-H$]
  wa8bxn v1.16, unterstützt BID und H-Adr.
- [MBL-5.14-H$]
  wa7mbl v5.12, unterstützt BID und H-Adr.
- [4RE-2.3-MH$]
  aa4re v2.3, unterstützt MID, BID, und H-Adr.
- [THEBOX-1.9c1-DHM$]
  Diebox v1.9c1, unterstützt MID, H-Adr. und 8stellige Rubriknamen
- [DP-4.10-AB1DFHMR$]
  DigiPoint v4.10, F6FBB & DieBox Optionen werden unterstützt.

28.3.2. Regeln für den Verbindungsaufbau

Sende den SID in der ersten Zeile nach dem Connect. Beantworte den SID (wenn er empfangen wurde) mit einem kurzen Kommandoprompt ">".

Wenn ein SID beim Connect empfangen wurde, antworte mit dem SID der eigenen Box.

Das verwendete Protokoll ergibt sich aus der Schnittmenge der SIDs der beiden Forward-Partner.

28.3.3. Das SEND-Kommando

Mit diesem Befehl wird eine Nachricht gesendet.
Sx TO [@ BBS[.LOC]] [< FROM] [$BID/MID]

x kann A, B, T oder P sein. Wenn x fehlt, so wird P angenommen, falls TO ein Rufzeichen ist, ansonsten wird ein B angenommen. Das Dollarzeichen $ ist kein Teil der MID, es kennzeichnet das Feld. Zwischen dem "$"-Zeichen und der BID darf kein Leerzeichen sein. Die Leerzeichen rund um den Klammeraffen @ können weggelassen werden. Beim User S&F muss das FROM-Feld mit dem Rufzeichen des Forwardpartners übereinstimmen. ST-Mails werden nicht angenommen.

28.3.4. Die OK/NO/REJ-Meldung

Die andere Box wird als Antwort auf das SEND-Kommando eine OK/NO/REJ-Meldung erzeugen, welche möglicherweise von irgendeinem Text gefolgt wird. Wenn die Antwort NO oder REJ ist, so wird sie von einem Prompt gefolgt. Wenn die Antwort OK ist, so muss der Forward-Partner (also jener, der SEND gesendet hat) die Mail forwarden. Normalerweise wird NO nur gesendet, wenn versucht wurde, eine Mail weiterzuleiten, deren BID bereits dem Empfangssystem bekannt war. REJ wird dann gesendet, wenn die Empfangsbox wegen eines Routingfehlers die Mail nicht annimmt. Eine solche Mail wird dann in die Datei *trace/sfhold.bcm* eingetragen. Eine Abfrage ist mit dem Befehl SFHOLD möglich.

Von den Meldungen "NO", "REJ" und "OK" muss jeweils nur der erste Buchstabe ("N", "R", "O") angegeben werden.

Zur Kompatibilität mit W0RLI/DPBox werden noch folgende Antworten unterstützt:

- LATER (W0RLI): Wie NO implementiert (Bedeutung: Empfange Mail bereits)
- HOLD (FBB): Wie OK (Nachricht wird Sysop vorgelegt)
- ERROR (FBB): Wie NO (Fehler im SEND-Befehl)

28.3.5. Die Übertragung der Nachricht

Wenn mit OK geantwortet wurde, sendet die Nachbarbox die Nachricht. Die erste Zeile der Nachricht ist der Nachrichten-Titel. Die weiteren Zeilen sind der Text der Nachricht. Die Nachricht wird durch <CR><Ctrl-Z><CR> beendet, von der empfangenden Mailbox wird durch ein Linefeed und das Prompt (">") der Empfang bestätigt. Darauf wird von der sendenden Box eine weitere Nachricht angeboten.

28.3.6. Richtungsumkehr

Falls keine Nachrichten mehr zum Forward bereitliegen, wird "F>" statt ">" als Prompt gesendet. Damit tauschen sendende Box und empfangende Box ihre Rolle.

28.3.7. Beendigung des Forward

Wenn auch diese Box keine Nachricht mehr anzubieten hat, beendet sie das Forward durch das Senden der Zeichenfolge "***done".

28.4. Der CHECK-Befehl

Um Kompatibilität zu den DieBox-Systemen zu gewährleisten, wurde zusätzlich zum Befehl DIR NEWS der Befehl CHECK eingebaut. Dieser Befehl liest aus der Datei *check.bcm*, die chronologisch aufgebaut ist und dasselbe Format hat, wie die Board-Listen *list.bcm* in den jeweiligen Verzeichnissen der Boards.

Die Datei *check.bcm* wird neu erzeugt, wenn sie fehlt oder fehlerhaft ist. Dazu werden von allen Boards die Dateien *list.bcm* aneinandergereiht. Sind die Listen alle eingelesen, so wird die Datei lexikalisch sortiert. Dieser Vorgang kann nicht im RAM, sondern muss auf der Festplatte erfolgen, da die Datei sehr groß werden kann (bei 30000 Mails ca. 4 MB). Der Sortiervorgang kann deshalb sehr lange dauern (je nach Rechner/Datenbestand) und erfolgt nur, wenn es unbedingt notwendig ist. Ein absichtliches Erzeugen von *check.bcm* ist mit dem Befehl REORG C möglich.

Bei der Ausgabe des Befehls CHECK wird vom Ende zum Anfang hin gelistet. Da zum Zeitpunkt der Auflistung die Nummer einer Mail innerhalb eines Boards nicht bekannt ist, wird die Datei *checknum.bcm* angelegt. In dieser Datei ist zu jeder in *check.bcm* aufgelisteten Mail die Position der Mail innerhalb ihres Boards abgespeichert.

28.5. Multitasking-Struktur

Über die Mailboxsoftware wird ein Task-Scheduler gelegt, der folgende Eigenschaften hat:

- Dynamisches Erzeugen und Löschen von Tasks. Eine Task kann an jeder beliebigen Stelle im Programm eröffnet werden (beliebige Schachtelungstiefe). Eine Task wird wieder gelöscht, wenn die zugehörige Prozedur beendet ist oder ein kill-Aufruf erfolgt (z.B. bei Disconnect).
- Unterbrechung des Ablaufs nur auf Anforderung (non-preemptive scheduling), d.h. der laufende Code wird nie an unvorhersehbaren Stellen unterbrochen. Dadurch braucht die gesamte Software nicht konsequent reentrant zu sein. Ausnahme: Bei allen Ausgaben muss mit einer Unterbrechung des Ablaufs gerechnet werden.
- Task-Kontrollblock: Alle vom Kontext abhängigen globalen Variablen werden im TCB untergebracht. Zu diesem wird ein stets gültiger Pointer bereitgehalten, über den alle TCB-Variablen referiert werden.

28.6. Reguläre Ausdrücke

"Reguläre Ausdrücke" sind Zeichen, mit denen man einen Suchtext anpassen kann. In der OpenBCM können sie bei Suchtexten bei den Befehlen CHECK, DIR, LIST, CONVERT, allen Log-Ausgaben von Trace-Dateien und vom Sysop in den Dateien *convert.bcm* und *reject.bcm* angewendet werden.

Zeichen	Beschreibung
^	Ein Circumflex am Anfang einer Zeichenkette findet den Anfang einer Zeile.
$	Ein Dollar-Zeichen am Ende einer Zeichenkette findet das Ende einer Zeile.
.	Ein Punkt findet jedes beliebige Zeichen.
*	Ein Stern nach einer Zeichenkette findet jede beliebige Anzahl an Vorkommen der Zeichenkette, gefolgt von beliebigen Zeichen (auch keine Zeichen). Beispiel: "bo*" findet "bot", "bo" und "boo" usw., aber nicht "b".
+	Ein Pluszeichen nach einer Zeichenkette findet jede beliebige Anzahl an Vorkommen der Zeichenkette, gefolgt von beliebigen Zeichen, aber nicht

```
        keine Zeichen.
        Beispiel: "bo+" findet "boo" und "booo", aber nicht "bo" und "be".
 \      Ein umgekehrter Schrägstrich vor einem Platzhalterzeichen bedeutet,
        dass Zeichen wörtlich und nicht als Platzhalterzeichen zu verwenden.
        Beispiel: "\^" findet "^" und sucht nicht nach dem Anfang einer Zeile.
 [ ]    Zeichen in Klammern finden jedes einzelne Zeichen, dass in der Klammer
        vorkommt, aber kein anderes.
        Beispiel: [bot] findet b, o oder t.
 [^]    Ein Circumflex am Anfang einer Zeichenkette in Klammern bedeutet NICHT.
        Beispiel: [^bot] findet alle Zeichen außer b, o oder t.
 [-]    Ein Bindestrich in Klammern zeigt einen Bereich von Buchstaben an.
        Beispiel: [b-o] findet jedes Zeichen von b bis o.
```

Geschweifte Klammern, wie sie bei anderen Implementierungen von "regulären Ausdrücken" üblich sind, sind in der OpenBCM nicht möglich.

Weitere Beispiele:

```
 ^[WE][WU]$     findet als Suchtext "WW", "WU", "EW" oder "EU".
 ^S\:           findet "S:" nur am Anfang vom Suchtext.
```

29. Extended White Pages Protocol (WPROT) - Spezifikation V1.0/Rev.2

Dieses Kapitel enthält eine Beschreibung des WPROT Protokolls, einem Verfahren für den Austausch von "White Page"-Informationen zwischen Packet-Radio Mailboxen. WPROT basiert auf den älteren Standards von W0RLI und F6FBB, die allgemein als "WP"-Verfahren bekannt sind. Diese Spezifikation wurde ursprünglich von OE3DZW entwickelt und greift auch Ideen zum Routing-Informationsaustausch von DL8HBS auf.

29.1. Funktionsübersicht

WPROT ist...

- hocheffizient
- bietet End-to-End-Fehlerkontrolle
- hochflexibel
- skalierbar
- Sicherung der Datenübertragung durch Checksumme

29.2. Kompatibilität

WPROT ist abwärtskompatibel zum WP-Standard von W0RLI und zum E/M-System wie es von Diebox genutzt wird. /G und /I (guess and info) Nachrichten von WP werden allerdings nicht unterstützt.

29.3. Wieso ein neues Protokoll?

Das originale WP Protokoll unterstützt nur den Austausch von Heimatmailbox-Informationen. Der Austausch dieser Daten ist jedoch unsicher (jeder User kann irgendwelche WP Nachrichten absenden), der Zeitstempel beinhaltet nicht den Zeitpunkt des Tages und es ist nicht kompatibel mit dem E/M-System von Diebox, dass immer noch von zahlreichen anderen Boxsystemen genutzt wird. Die Y2K-Kompatibilität ist bei älteren Umsetzungen des WP Protokolls nicht gegeben (Datum im "990407"-Format).

Das E/M-Protokoll von DieBox bietet ebenfalls Nachteile: es ist sehr uneffektiv, denn nur eine Information kann pro Mail versendet werden. Es unterstützt zwar Remote-Erase (Fernlöschung) von Rubriknachrichten, dies ist technisch jedoch nicht bis ins letzte Detail gut implementiert - somit konnte dies leider in der Vergangenheit von Löschsysops massiv missbraucht werden.

WPROT ist hingegen mit E/M und WP kompatibel, nutzt die Vorteile von beiden Protokollen und bietet eine zusätzliche Funktionalität.

29.4. Aktivierung von WPROT

Der Sysop hat lediglich sicherzustellen, das WPROT Daten zu WPROT-kompatiblen Nachbarboxen gesendet werden. In OpenBCM geschieht dies, indem im Forwardabschnitt zur Nachbarmailbox das Schlüsselwort $WP in die Datei *fwd.bcm* eingetragen wird. Es ist darauf zu achten, dass dann das Schlüsselwort $EM nicht ebenfalls in diesem Forwardabschnitt steht!

29.5. System Identifier (SID)

Systeme, die das WPROT Verfahren unterstützen, haben den Buchstaben "W" in ihrer SID. Durch diesen Buchstaben weiß die Mailbox, dass WPROT Daten anstelle von WP Daten unterstützt werden.

Hinweis: Ob E/M Daten der Diebox akzeptiert werden oder nicht, wird durch den Buchstaben "D" in der SID definiert.

Beispiel einer SID von OpenBCM:
[OpenBCM-1.04-AB1D1FHMRW$]

29.6. Format der WPROT Nachrichten

WPROT Daten werden an W@<Nachbar-Mailbox>, also z.B. an W@DB0ROF adressiert. Die Rubrik W wird benutzt, da es sich um eine einbuchstabige Rubrik handeln und somit für den allgemeinen Benutzer versteckt geführt wird (dies ist bei OpenBCM, DPBOX und auch DieBox so).

- Mailtitel: "WPROT Update" (sollte beim Empfang nicht überprüft werden)
- Typ der Mail: Usermail (Typ "P"(ersonal))
- Absender: das eigene Mailboxrufzeichen
- Empfänger: Nachbar-Mailbox
- Falls Lifetime unterstützt wird: 1 Tag (sollte beim Empfang nicht überprüft werden)
- Adressierung der Mail: W@<Nachbar-Mailbox>

Beim Mailempfang sollen nur Mails vom direkten Forwardpartner ausgewertet werden, die ein "W" in ihrer SID anbieten. Dies soll als Sicherheit dienen, da man nur bei direkten Forwardpartnern eine sichere Kommunikation voraussetzen kann.

Eine Nachricht kann willkürlich beliebig lang sein. Sie besteht aus individuellen Feldern, die jeweils eine Zeile lang sind. Die einzelnen Zeilen werden durch CR (wie in Packet Radio üblich) getrennt. Alle Zeichen sind 8bit groß. Es werden Standard ASCII-Zeichen genutzt.

(Hinweis: Nur 7bit Zeichen sind in ASCII definiert, alle Zeichen > 0x7f sind undefiniert).

Die Länge von einigen Feldern/Zeilen (z.B. Name, QTH) ist bei einigen Systemen limitiert. Falls ein Eintrag die lokalen Limits überschreitet, soll dieser Eintrag verworfen werden.

Beispielnachricht:

```
------
W < OE3DZW @OE1XAB  $F29OE3DZW005
------
WP Update
R:990407/0944z @:OE3DZW.#OE3.AUT.EU [BayCom-Mailbox] Bcm1.40P

D7 V 10
68 B OE3DZW.#OE3.AUT.EU 10 0 DL8MBT AX25 OE3DZW-8 923478289 1
5C M OE3DZW OE3DZW.#OE3.AUT.EU 749OE3DZW004 923471053 OE3DZW 1 "Dietmar" ? "?"
00 E WW OE3DZW F29OE3DZW007 F29OE3DZW006 OE3DZW 1 "#E"
```

29.7. Format der WPROT Zeilen

Für jede Information wird genau eine Zeile genutzt. Eine Information hat folgendes globales Format:

```
    <checksum> <type> <data>
```

Die <checksum> ist eine einfache Byte-Checksumme, die bei 00 beginnt und über <flag> bis <data> gebildet wird. Sie beinhaltet keine Leerzeichen vor <flag> und kein CR nach <data>. Die Checksumme besteht aus einem zweistelligen Hex-Wert, der Grossbuchstaben verwendet (z.B. "4F").

Die <type> Deklaration kann mehrere bis maximal 10 Zeichen lang sein, normalerweise ist sie jedoch nur ein Zeichen lang.

Beispielzeile:

```
48 M OE3DZW OE3DZW.#OE3.AUT.EU F29OE3DZW004 919094319 OE3DZW 1 "Dietmar" ? "?"
^checksum
  ^type
    ^data (abhängig von <type>)
```

Generelle Regeln:

- Zeilen werden durch "<CR>" getrennt, andere Zeilenumbrüche werden nur beim Empfang akzeptiert.
- <type> werden durch Leerzeichen "0x20" getrennt; Tabulatoren oder andere White Space Zeichen sind nicht erlaubt (aber werden evtl. beim Empfang akzeptiert)
- Zeichen sind im allgemeinen in Grossbuchstaben zu benutzen, Kleinbuchstaben dürfen nur in bestimmten Fällen bei <data> angewendet werden (z.B. bei Namen, QTH)
- Falls ein Datenparameter Leerzeichen enthält/enthalten darf, müssen Anführungszeichen (z.B. "van Dark") benutzt werden.
- Falls ein Datenparamter unbekannt ist, soll ein "?" benutzt werden. Parameter dürfen nicht weggelassen werden.
- Zeitstempel sind generell im ANSI-Format (signed 32bit Integer) zu führen, 0 ist als der 1.1.1970 00:00 GMT definiert
- Alle Felder eines Eintrages sind zwingend auszufüllen

29.8. Zeilentypen

Bislang sind folgende Typen <type> definiert:

- V Version
- B BBS-Info
- E Remote-Erase
- M MyBBS-Info (Heimatmailbox)

- R Routing-Info

Die Typen V und M müssen mindestens von jedem System, das WPROT nutzt, unterstützt werden. Die anderen Typen dürfen beim Empfang ignoriert werden, besser ist natürlich, wenn diese von alle WPROT-fähigen Mailboxsystemen verstanden werden.

29.8.1. Typ V

Dieser Typ wird genutzt, um die Version des unterstützten WPROT-Protokolls mitzuteilen.

Format:
<version>

<version> ist eine 32bit Integer-Zahl vom Typ "unsigned". Eine Zahl von z.B. "10" steht für Version 1.0. Eine Zeile vom Typ V sollte immer am Anfang einer WPROT Nachricht stehen.

Falls die empfangene Versionsnummer höher als die höchste bekannte Versionsnummer vom lokalen eigenen System ist, soll die WPROT Nachricht nicht ausgewertet werden.

29.8.2. Typ B

Dieser Typ wird genutzt, um Mailbox-Informationen auszutauschen.

Format:
<h-Adr> <sysop call> <com prot> <hw-adr> <timestamp> <hop>

- <h-Adr> Hierarchische Adresse der Mailbox
- <sysop call> Rufzeichen des Sysops (ohne Name/MYBBS)
- <com prot> Protokolltyp, der für die Funkkommunikation verwendet wird, bislang sind folgende Typen definiert:
  - AX25.... Amateur-Radio Wireless Protokoll
  - TELNET.. RAW Protokoll
  - TCP/IP TCP/IP V4 Verbindung
  - ?....... Unbekanntes Protokoll
- <hw-adr> Abhängig vom Protokolltyp <com prot>:
  - AX25: AX25-Adresse mit/ohne via-Digis (z.B. "DB0AAB-8")
  - TELNET: IP-Nr. und Port, z.B. "db0aab.ampr.org:4719"
- <timestamp> ANSI-Zeitstempel vom Zeitpunkt, als die Nachricht erzeugt wurde
- <hops> Anzahl an Forward-Hops. Beim ersten Forward wird "hops" auf "1" gesetzt.

29.8.3. Typ M

Dieser Typ wird genutzt, um Heimat-BBS-Informationen auszutauschen.

Format:
<call> <home bbs> <bid> <timestamp> <origin bbs> <hops> <name> <zip> <qth>

- <call> Rufzeichen des Users, dessen Heimatmailbox gemeldet wird
- <home bbs> Volle hierarchische Adresse der Heimatmailbox. Darf keine SSIDs beinhalten! Der hierarchische Teil der Adresse kann u. U. weggelassen werden, der linkste Teil der Adresse muss ein Mailboxrufzeichen sein. Beispiele:

- OE3XSR.#OE3.AUT.EU   -> gültig
- OE3XSR-8.#OE3.AUT.EU -> ungültig
- OE3XSR-9   -> ungültig
- BBS.OE3XSR.#OE3.AUT.EU-> ungültig
- OE3XSR   -> gültig

* <bid> Es erscheint erstmal dumm, hier eine BID zu nutzen, aber: DieBox benutzt eine BID für jede MYBBS-Information, somit kann diese Information an dieser Stelle weitergegeben werden. Falls die BID unbekannt oder nicht verfügbar ist, muss ein "?" an dieser Stelle genutzt werden. Hinweis: BIDs werden bei einigen Systemen nach Groß-/Kleinschrift unterschieden, deshalb darf an dieser Stelle die Groß-/Kleinschrift der BID niemals verändert werden!
* <timestamp> Zeitstempel der Heimatmailbox-Information
* <origin bbs> Rufzeichen der Mailbox, von der die ursprüngliche MYBBS Information herrührt, ohne Hierarchische Adresse und ohne SSID.
* <hops> Hops Zähler: Anzahl an Forwardboxen. 1 beim ersten Forward.
* <name> Name des Users, z.B. "Dietmar" oder "Hans-Jürgen"
* <zip> ZIP-Code (=Postleitzahl) des QTHs, z.B. "CH-3861"
* <qth> Heimat-QTH des Users, z.B. "Düsseldorf"

Beim Empfang macht es keinen Sinn, die BID in die lokale BID-Datenbank aufzunehmen!

29.8.4. Typ E

Dieser Typ wird genutzt, um Remote-Erase(=Fernlösch)-Informationen auszutauschen.

Format:

<flood> <bbs> <bid> <erase_bid> <call> <hops> <comment>

* <flood> Verteiler der Nachricht, die gelöscht werden soll, darf länger als 6 Zeichen sein!
* <bbs> Rufzeichen der Mailbox, wo die Erase-Mail generiert wurde
* <bid> Es erscheint erstmal dumm, hier eine BID zu nutzen, aber: DieBox benutzt eine BID für jede MYBBS-Information, somit kann diese Information an dieser Stelle weitergegeben werden. Falls die BID unbekannt oder nicht verfügbar ist, muss ein "?" an dieser Stelle genutzt werden. Hinweis: BIDs werden bei einigen Systemen nach Groß-/Kleinschrift unterschieden, deshalb darf an dieser Stelle die Groß-/Kleinschrift der BID niemals verändert werden!
* <erase_bid> BID der Nachricht, die gelöscht werden soll
* <hops> Anzahl an Hops
* <comment> Grund, wieso die Nachricht gelöscht werden soll, z.B. "#K" oder "?"

Beispiel:

Die originale Nachricht:

```
------------
TEST < OE3DZW @WW                    $F29OE3DZW00
------------
R:990215/1734z @:OE3DZW.#OE3.AUT.EU [BayCom-Mailbox] Bcm1.40mg
From: OE3DZW @ OE3DZW.#OE3.AUT.EU
To:   TEST @ WW
TEST < OE3DZW @WW $F29OE3DZW006

(..)
```

wird durch

```
2F E WW OE3DZW F29OE3DZW007 F29OE3DZW006 OE3DZW 1 "#E"
```

gelöscht.

29.8.5. Typ R

Dieser Typ wird genutzt, um Routing-Informationen auszutauschen.

Format:

```
<call> <version> <timestamp> <hops> <qual>
```

- <call> Rufzeichen
- <version> Version
- <timestamp> Zeitstempel der MYBBS
- <hops> Anzahl an Hops
- <qual> Qualität der Forwardgeschwindigkeit (Sekunden für das Versenden von 100 kBytes Daten, als ASCII Wert für einen Zahlenwert vom Typ "unsigned long")

Beispiel:

```
F3 R DB0SIF 10 950307915 2 16404
```

29.9. Berechnung der Routingquality für den Typ R

Die Berechnung der Qualität einer Forwardroute kann durch Betrachtung der gesendeten Daten an eine Nachbarbox erfolgen, denn nur dadurch kann man verlässliche Informationen darüber gewinnen, ohne das spezielle Routingdaten ausgetauscht werden müssen.

Die einzige Möglichkeit, eine Zeit messen kann, ohne das ein Idlezustand das Ergebnis verfälscht, ist die Zeitdauer zwischen dem Startzeitpunkt, wo eine Mail gesendet wird (nicht der Start des Proposalvorschlags!) und der Empfangsbestätigung. Diese Zeitdauer kann dann auf einen Wert in Sekunden pro 100 kBytes umgerechnet werden.

Um die erzielte Bandbreite eine komprimierten FBB Forwards wiederzuspiegeln, wird nicht die Anzahl an gesendeten Daten zur Berechnung herangezogen, sondern das tatsächliche unkomprimierte Datenvolumen. Dadurch erhalten komprimierte Forward-Verbindungen einen besseren Qualitätswert als unkomprimierte bei gleicher Hardware-Bandbreite. Dies ist auch so beabsichtigt.

Kleine Datenmengen von weniger als 512 Bytes werden nicht zur Berechnung herangezogen, da hier der Einfluss anderer Faktoren eine vernünftige Berechnung beeinflusst. Wenn die letzte Messung mehr als 60 Minuten her ist, werden jedoch auch solche kleinen Datenmengen zur Berechnung herangezogen, bis der nächste größer als 512 Bytes große Datenblock gesendet wurde.

Die Qualität einer Forwardverbindung ist der Mittelwert aller Messungen in den letzten 5 Stunden. Dies verhindert allzu starke Sprünge in den Qualitätswerten.

Der letzte Mittelwert wird beibehalten, auch wenn in den letzten 5 Stunden keine neue Forwardverbindung mit der Partnermailbox stattgefunden hat.

Der Wert altert um jeweils 20% seines Anfangswertes pro Stunden und mindestens um eins pro Stunde. Nach einem Tag wird der Wert auf den Defaultwert einer unbekannten Forwardverbindung gesetzt (= 32767). Nach drei Tagen wird der Qualitätswert komplett aus der Forwardtabelle gestrichen.

Der maximale Qualitätswert ist 1 (d.h. 1 Sekunden für das Senden von 100 kBytes Daten), der minimale Qualitätswert ist 32767 (d.h. 32767 Sekunden für das Senden von 100 kBytes Daten).

Falls eine Forwardverbindung eine nicht-interaktive Methode verwendet (z.B. ein Dateiforward), wird die Qualität fest auf 16383 gesetzt. Eine Berechnung ist für solche Forwardverbindungen nicht möglich.

Bei jeder Erhöhung des Hop Zählers wird der Qualitätswert jeweils fix um den Wert 100 plus 10% des ursprünglichen Qualitätswertes erhöht.

Später könnte auch eine Erhöhung des Qualitätwertes um die Anzahl an Mails, die noch in der Forwardschlange zu der Partnermailbox stehen, eingeführt werden. Dies ist aber im Moment nicht vorgesehen.

29.10. Routing der WPROT Nachrichten

Einige Einträge (Nachrichten mit Hop Zähler) dürfen zu anderen Mailboxen weitergeleitet werden, falls

- der Zeitstempel des Eintrags neuer ist, als der zuletzt empfangene (bei M und B Informationen) (*)
- die Erase-BID neuer ist (E Informationen)
- die Nachricht nicht älter als 1 Monat ist
- Die Nachricht soll nicht an den Absender zurückgeschickt werden
- Der Hop Zähler soll jedes Mal um eins erhöht werden
- Die Information soll nicht mehr weitergeleitet werden, wenn der Hop Zähler einen Wert von 50 erreicht hat.

(*) falls ein vorheriger Eintrag mit gleichen Daten bereits empfangen wurde, soll ein weiteres Update in einem speziellen Zeitfenster unterdrückt werden (z.B. B Informationen werden nur geupdatet wenn sie sich verändert haben oder einmal pro Monat)

Falls die lokale Mailbox einen speziellen Typ nicht unterstützt, werden Nachrichten ggf. nicht geforwardet. Die Typen E und B müssen nicht unbedingt unterstützt werden, M muss hingegen immer unterstützt werden.

29.11. Gemischte WP/E&M/WPROT-Systeme

Bei gemischten WP/E&M/WPROT-Systemen ist folgender Datenaustausch möglich:

```
WP   <-> WPROT (nur Typ M)
E/M  <-> WPROT (Typen M und E)
WP   <-> E/M   (nur Typ M)
```

Bei der Konvertierung von WP nach WPROT oder E/M wird der Zeitstempel vom Typ YYMMDD (z.B.: 990215) ins ANSI-Zeitformat gewandelt: Sec: 00 Min: 00 Hours: 00 (GMT), -1 day (= -86400s) (!!)

Bei der Konvertierung von WPROT oder E/M nach WP wird der Zeitstempel durch Abschneiden der Stunden/Minuten gewandelt.

29.12. Sicherheit

Nur Nachrichten, die direkt vom Forwardpartner empfangen wurden, werden ausgewertet.

30. OpenBCM Skriptsprache

Sowohl in der Windows, als auch in der Linux-Version ist eine Makrosprache implementiert (BCSL = BayCom Script Language). In der DOS-Version ist diese Makrosprache nicht vorgesehen (zu wenig Speicher).

Die Makrosprache ist bis jetzt noch eine Baustelle.

!!! Vorsicht. Diese Beschreibung ist unvollständig und in Teilen falsch! Auch die Software ist noch unvollständig und ungetestet!!!

Dieses Kapitel dient lediglich der Sammlung von Ideen.

30.1. Vorwort

Es wird oft gewünscht, in eine vorhandene Mailbox zusätzliche Funktionen einzubinden. Eine gewisse Möglichkeit ist durch die Einbindung von Run-Utilities gegeben. Diese sind allerdings sehr unflexibel und können nur stark eingeschränkt auf die Datenbestände der Mailbox zugreifen.

Um diese Lücke zu schließen, wurde eine Skriptsprache entwickelt, mit der kleine Zusatzfunktionen realisiert werden können, für die das Mailboxprogramm nicht verändert werden muss. Was dabei herausgekommen ist, wird jedem Informatiker, der sich bereits mit formalen Sprachen beschäftigt hat, die Haare zu Berge stehen lassen. Beruhigend ist jedoch die Tatsache, dass es relativ viele Skriptsprachen gibt, die ebenso wirr strukturiert sind und dabei einen weit größeren Verbreitungsgrat haben als BCSL.

30.2. Starten eines Skriptes

Ein Skript kann zu Testzwecken vom Sysop per Befehl gestartet werden. Dies geht mit "macro <scriptname>". Die Skripte stehen in einem Verzeichnis namens *macro*. Jedes Skript wird in einem eigenen File abgelegt. Der Name des Skriptes ist der Dateiname (ohne Endung, bzw. mit Endung ist auch diese Bestandteil des Scriptnamens) Im praktischen Betrieb erfolgt die Abarbeitung von Skripten ereignisgesteuert. Immer wenn ein Ereignis unter bestimmten Randbedingungen eintritt, kann ein bestimmtes Skript ausgeführt werden. Die Verbindung zwischen Ereignissen und Skriptnamen wird in einem File namens *macro.bcm* festgelegt.

Dieses File hat Einträge im folgenden Format:

```
<event> [<call> | <command> | <wildcard>]: <macro>
```

Folgendes Beispiel zeigt grob die Möglichkeiten:

```
login *: loginout
login dg3rbu: rbu
accept db0aab-3: aab3
cmd a*ktuell: akt
cmd pu*rge: disabled
scmd fwdadd: fwdadd
beforecmd v*ersion: b
beforecmd *:
aftercmd *:
mailfm dh3vy:
deletemail mailto *:
unknown db0kfb*: nobox
```

In der ersten Spalte wird das Ereignis benannt, das zur Ausführung des Skriptes führen wird. Hinter dieser Angabe erfolgt ein Parameter, der das Ereignis entweder konkretisiert oder einschränkt. Abkürzungen mit * sind meist möglich, die alleinige Angabe von * ist grundsätzlich zutreffend und ist gleichwertig mit

dem Weglassen dieses Parameters, d.h. "login *: script" und "login: script" verhalten sich gleich.

Hier gibt es folgende Möglichkeiten:

login [<call>]: Nach jedem Login wird dieses Makro ausgeführt. Der Zeitpunkt der Ausführung ist, nachdem alle Loginformalitäten erledigt sind aber noch bevor das erste Kommando (siehe "alter command") ausgeführt wurde. Durch einen Parameter kann das Rufzeichen des Logins eingeschränkt werden. Mittels * ist das Rufzeichen abzukürzen. Denkbar wäre z.B. ein Konstrukt: "login oe*: oegruss", wenn im Skript "oegruss" alle Freunde aus unserem südöstlichen Nachbarland besonders herzlich begrüßt werden.

accept <call>: Hier wird ein Connect von außen unter dem angegebenen Rufzeichen entgegengenommen. Der Connect wird entgegengenommen, die Kommunikation mit dem neu ankommenden Benutzer wird direkt an das angegebene Skript übergeben.

Wird das Skript beendet, so erfolgt ein Disconnect. Bei Logins über Telnet oder am Bildschirm kann das Rufzeichen als zweiter Parameter beim Login angegeben werden. An dieser Stelle ist eine Abkürzung mit * nicht zulässig.

cmd <command>: Fügt ein neues Kommando der Boxoberfläche hinzu. Das Kommando kann mittels * in einen verbindlichen und einen optionalen Teil zerfallen.

Beispiel:
new*command bedeutet, dass ein neuer Befehl "newcommand" eingefügt wird, wobei "new" in jedem Fall eingegeben werden muss, der Rest kann entfallen. So eingeführte Befehle können eingebaute Befehle der Mailbox überschreiben, diese sind dann mit dem Original-Syntax nicht mehr ausführbar.

Durch Voranstellen eines "-"Zeichen wird jedoch die Ausführung des eingebauten Befehles erzwungen, so dass ein tatsächliches Abdecken eines Befehles nicht möglich ist (dies ist kein Bug sondern Absicht).

scmd <command>: wie "cmd", jedoch sind so erzeugte Kommandos nur im Sysop-Modus ausführbar.

beforecmd <cmd> und **aftercmd <cmd>**: Hier referenzierte Skripte werden vor bzw. nach der Ausführung des mit <cmd> referenzierten Kommandos ausgeführt.

Die Abkürzung des Kommandos muss genauso aufgeschrieben werden, wie dieses von der Box interpretiert wird.

mailfrom <call> und **mailto <call>**: Dieses Skript wird aufgerufen, wenn eine neue Nachricht in der Box eintrifft. Über den Parameter kann ggf. eingeschränkt werden, zu wem (bzw. in welche Rubrik) oder von wem eine Nachricht eingespeichert wurde. Beim Forwarding ist dabei das tatsächliche Absenderrufzeichen der Mail relevant, nicht das Loginrufzeichen der Forwardbox. Wird kein Parameter oder * angegeben, sind die Ereignisse "mailto" und "mailfrom" identisch.

unknown <adr>: Das referenzierte Skript wird ausgeführt, wenn eine Nachricht mit einer unbekannten Forwardadresse eintrifft. Bei der Adresse ist eine Abkürzung mit * möglich.

30.3. Syntaxelemente der Skriptsprache

Die BCM-Makroskriptsprache ist eine frei erfundene Programmiersprache, die zwar sehr viele Gemeinsamkeiten mit anderen Sprachen hat, zu keiner jedoch identisch ist. Der Kritiker mag dies schimpfend bemängeln, wird es aber auch nicht mehr ändern können. Als Begründung, warum die Sprache so aussieht und nicht anders

kann man lediglich anführen, dass sie eben so geworden ist. Eine belastbare technische Begründung gibt es nicht. Wesentlich ist die Tatsache, dass es nicht möglich ist, komplexe Programme in BCSL zu schreiben.

Um keine falschen Hoffnungen zu wecken, seien hier zunächst einmal die Schwächen von BCSL aufgezählt:

- mit BCSL sind nur überschaubare, oberflächliche Problemstellungen lösbar.
- Es existieren keine strukturierten oder indizierten Datentypen.
- Unterprogramme sind nur mit gewissen Einschränkungen möglich.
- Durch textuelle Interpretation ist die Ausführung der Skripten ausgesprochen langsam.
- Es ist nur Integerarithmetik mit den Grundrechenarten vorhanden.
- Durch relativ hohen Speicherbedarf ist die Funktion in der DOS-Version normalerweise gar nicht, in Sonderfällen nur eingeschränkt nutzbar.

Jedes Skript wird durch die Schlüsselwörter "begin" und "end" eingeklammert. Auch Kontrollstrukturen ("if" und "loop") werden gezielt durch "endif" und "endloop" beendet. Jedes Statement wird mit ";" abgeschlossen. Es gibt einen Unterschied zwischen Prozeduren und Funktionen. Funktionen haben einen Rückgabewert, dieser muss auch weiterverarbeitet werden (z.B. in einem Klammerausdruck oder einer Zuweisung). Prozeduren haben keinen Rückgabewert und können nicht zugewiesen werden. Das Verhalten in dieser Beziehung ist ähnlich Pascal. Innerhalb einer Blockstruktur (zwischen "begin" und "end" oder sinngemäß innerhalb einer Kontrollstruktur "if" oder "loop") wird eine Serie von Statements ausgeführt, die jeweils durch ";" abgeschlossen werden.

30.4. Prinzipieller Aufbau eines Skripts

Der prinzipielle Aufbau eines Skripts erfolgt nach folgendem Muster:

```
begin <script-Block>
    if( Bedingung )
         <script-Block>
    else
         <script-Block>
    endif;
    loop
         <script-Block>
         exit_if( Bedingung );
         <script-Block>
    endloop;
    <script-Block>
end.
```

Beispiel:

```
// Skript zur Ausgabe des Aktuelltextes
begin
    ftime=filetime("msg/aktuell.dl");
    if(ftime!=0) put("Aktuell vom " & datestr(ftime) & " " & timestr(ftime)
         & "\n");
      cmd("rt msg/aktuell.dl");
    else put("Kein aktueller Text geladen.\n");
    endif;
end.
```

30.5. Kommentare

Im Skript dürfen an beliebigen Stellen Kommentare stehen. Die Konventionen entsprechen hier der Sprache C++:

- Die Zeichen /* und */ schließen einen Kommentar innerhalb einer Zeile ein. Hierbei ist keine Schachtelung zulässig.
- Das Zeichen // leitet einen Kommentar bis zum Ende der Zeile ein.

- Kommentare dürfen an beliebigen Stellen im Quelltext stehen, jedoch nicht innerhalb eines Bezeichners (Konstante, Variable, Schlüsselwort etc).

30.6. Variablen

Variablen werden implizit deklariert. Das heißt, dass eine Variable als deklariert und definiert gilt, sobald sie die linke Seite einer Zuweisung passiert hat. Eine Variable kann ohne Typdeklaration sowohl numerischen als auch alphanumerischen Inhalt annehmen, auch eine Zuweisung von einem Typ zum anderen ist zulässig. Eine leere (aber definierte) Variable kann durch x=""; erzeugt werden. Der Inhalt einer Variablen kann beliebig lang werden (Begrenzung durch die Ressourcen des Rechners, bei DOS 64kByte), numerische Werte reichen von -2^31 bis 2^31 (32 Bit Integer).

30.7. Konstanten

Im Programmtext werden Stringkonstanten in "Anführungszeichen" angegeben. Bei numerischen Konstanten ist sowohl eine Angabe mit als auch ohne Anführungszeichen zulässig. Innerhalb von Stringkonstanten können die C-üblichen Steuerzeichen verwendet werden, das sind:

- \n Newline, also Sprung an den Anfang der nächsten Zeile
- \a Klingelzeichen, also CTRL-G
- \" Anführungszeichen "
- \xhh Hexadezimale Angabe des Zeichencodes, der eingesetzt werden soll. Die Angabe wird stets 2stellig erwartet, z.B. \x00 oder \xff
  Beispiele: x=10; y="das ist ein String mit Zeilenende\n"; xx=27+y; ist gleichwertig zu xx="27"+y;

30.8. Reservierte Wörter

Die Schlüsselwörter von BCSL teilen sich auf in:

- Operatoren
- Strukturanweisungen
- interne Prozeduren
- interne Funktionen

Operatoren (Auflistung in der Reihenfolge der Priorität):
Bei allen arithmetischen Operationen werden leere Variablen, oder Variablen die keine Zahl beinhalten, zu 0 angenommen (eine solche Operation führt nicht zum Fehler). Bei logischen Operationen gilt der Wert 0 (oder keine Zahl) als "FALSE", jede Zahl ungleich 0 gilt als "TRUE". Zurückgegeben wird "0" für "FALSE" und "1" für "TRUE". Das Verhalten in dieser Beziehung ist gleich der Sprache C.

Für Variablen, die keine Zahlen enthalten (also Strings) sind nur die Operatoren "==", "!=" und "&" sinnvoll einsetzbar.

- Punktoperatoren:
  *   multipliziert numerische Ausdrücke
  /   dividiert numerische Ausdrücke
  mod Modulo-Operation (Rest der Division)
- Strichoperatoren:
  +   addiert numerische Ausdrücke
  -   subtrahiert numerische Ausdrücke
- Vergleichsoperatoren:
  >   größer als

<   kleiner als
>= größer oder gleich
<= kleiner oder gleich
== gleich (bei Strings: identisch, Groß-/Kleinschreibung irrelevant)
!= ungleich (auch für Strings anwendbar)

- logische Operatoren
  0   oder
  and logisches UND
  or logisches ODER
  xor exklusives ODER
- Stringverkettung:
  &   fügt Strings aneinander

30.9. Strukturanweisungen

- begin und end umklammern ein ganzes Skript.
- Jedes Skript wird durch die Folge begin <block> end; eingeschlossen.
- if, else und endif machen eine Verzweigung nach einer Bedingung möglich. Das Verhalten ist mit jeder anderen strukturierten Programmiersprache vergleichbar.
- Eine Schachtelung solcher Strukturen ist möglich und hat nicht das von C bekannte Problem der Uneindeutigkeit von "else", da jede Struktur mit "endif" eingeklammert wird.
  if(<Bedingung>) <block> endif; oder
  if(<Bedingung>) <block> else <block> endif;
  Der Ausdruck <Bedingung> wird wie in C numerisch ausgewertet. Jeder i numerische Wert ungleich 0 ist "true", 0 ist "false".
  Ein nichtnumerischer Wert wird ebenfalls als "false" interpretiert.
- loop, exit, exit_if, endloop: Dies sind die Elemente des Schleifenkonstruktes. Ein solches Konstrukt ist z.B. aus der Sprache ADA bekannt und ersetzt die verschiedenen Konstrukte wie "do ... while" oder "repeat ... until". Auch eine FOR-Schleife existiert hier nicht.
  loop <block> exit_if(<Bedingung>);
  <block> endloop;
  Der Block vor oder nach dem "exit_if"-Ausdruck kann jeweils auch entfallen, dadurch erfolgt die Überprüfung der Ausstiegsbedingung vor oder nach dem Schleifendurchlauf. Die Bedingung wird wie bei "if" ausgewertet. Das Schlüsselwort "exit" bewirkt einen harten Ausstieg aus der Schleife ohne Bedingung, dies wird jedoch wohl eher selten benötigt.

30.10. Interne Prozeduren

- abort(<reason>); bricht ein Makro hart ab.
- call(<macroname>[,<var1>,...]); ruft ein Unterprogramm auf. Dabei können Variablen für das Unterprogramm sichtbar gemacht werden.
- disconnect(<handle>); geht noch nicht.
- filewrite(<filename>,<data>,<mode>); schreibt eine Variable in ein File.
- oshell(<command>); führt einen shell-Befehl aus.
- put(<string>); gibt einen String zum Benutzer aus.
- sleep(<milliseconds>); suspendiert das Skript.
- trace(<level>,<string>); gibt einen Trace ins Syslog aus.
- cmd("e dl8mbt 1-"); führt ein Boxkommando aus. Ausgaben erfolgen zum Benutzer.

30.11. Interne Funktionen

- x=chr(65); * one ASCII-character erzeugt ein Zeichen gemäß angegebenem Code.
- x=cmd("v"); führt ein Kommando aus und weist die Ausgabe des Kommandos einer Variable zu.
- handle=connect("dl8mbt","db0aab"); geht noch nicht.
- x=datestr(gettime); wandelt ANSI-Zeit in lesbaren String um.
- line=fileline("filename",5); geht noch nicht.
- string=fileread("filename"); liest ein File ein und weist es einer Variable zu.
- size=filesize("filename"); gibt die Größe einer Datei zurück.
- t=filetime("filename"); gibt den Zeitstempel einer Datei als ANSI-Zeit zurück.
- s=getfname("dl8mbt",25); geht noch nicht.
- s=getline; geht noch nicht. s=getlist("dl8mbt"); geht noch nicht.
- s=getmacro("%o"); geht noch nicht.
- t=gettime; gibt die momentane ANSI-Zeit zurück.
- s=getuser("dl8mbt","forward"); gibt eine ALTER-Einstellung eines Users zurück. Folgende Einstellungen können abgefragt werden:
  check command echo fdelay fhold forward grep helplevel idir ilist iread lf lines mybbs name nopurge prompt pw pwline quota readlock reject rlimit speech status ttypw udir ulist uread
  ; geht noch nicht.
- boolean=getvalid; geht noch nicht.
- call=getvar("logincall"); gibt den Inhalt einer System-Variable zurück. Folgende Bezeichner können übergeben werden:
  logincall uplink txbytes rxbytes sysop logintime board dest src frombox at bid boxlist usermail lifetime lines bytes subject replyboard replynum taskid taskname created lastinput lastcmd
  ; geht noch nicht.
- i=len(string); gibt die Länge eines Strings zurück.
- a=lines(x); gibt die Anzahl der Zeilen in einem String zurück, geht noch nicht.
- x=oshell("ls -l"); or oshell("ls -l"); shell-Ausstieg. Zuweisung geht noch nicht.
- s=strline(x,5); gibt die 5-te Zeile eines Strings zurück, geht noch nicht.
- i=strpos(s,"test"); gibt die Position eines Substrings in einem String zurück.
- t=timestr(time); wandelt ANSI-Zeit in Zeitstring um
- s1=token(command,1); zerlegt einen String in einzelne Teile, geht noch nicht.
- x=val("A"); gibt den Zeichencode des ersten Zeichens im String zurück

Wichtig: Dieses Kapitel dient lediglich der Sammlung von Ideen!

31. DFWD - Spezifikation V0.1

Dieses Kapitel enthält eine Ideensammlung von OE3DZW zum DFWD-Konzept, und ist in den aktuellen OpenBCM-Mailbox-Versionen nicht implementiert! DFWD steht für "direct forward" stehen. Es sollen Usermails direkt von der Quellmailbox zur Zielmailbox geschickt werden, ohne das diese den Weg über andere Mailboxen nehmen.

31.1. Abstract

Direct forwarding (no relaying) between Amateur Radio BBSes as an alternative to Store & Forward used today.

31.2. Einführung

Today we have a very dense network of BBSes and digis. I takes seconds to leave a mail in a remote BBS, but can take hours..days to forward a personal mail from a local BBS to the same destination - if it arrives at all. The mails is forwarded and stored in several BBSes, at each hop there is a chance of:

- Misconfiguration by the local sysop
- Usage of old or buggy software
- Inconsequent routing due to the usage of different routing concepts
- Unavailability of BBS
- Congestion of a BBS due to overload (eg. big bulletins, usermail has to wait until fwded)

In this specifiation direct forward between two BBS is defined.

31.3. Wie es funktioniert

Which mails should be forwarded directly from A to B:

- Address: <valid_call> > <valid_call> @ <B>
- Mailtype P
- May originate in A (or not)

There is no technical reason why not to forward mails from or to no-calls (eg. servers). But servers are usually causing trouble, therefore leave them out now, they can be added later.

How to know that the remote B supports DFWD

- received WPROT-broadcast from that BBS

(or any similar method, eg. sysop-configuration)

That broadcast contains

- Flag that DFWD is accepted
- Connect Path to the remote BBS

How to connect to the remote BBS

A Flexnet-compatible network is assumed, it can be modelled as:

- call+active-ssid of BBS A
- uplink network (eg. local digi)
- fwd-call (ssid) of BBS B

Eg: BBS A knows that it needs to connect to OE1XLR to get access to the network (due to sysop configuration) and has a defined bbs-call and "fwdssid (ssid used for outgoing connects). BBS B has broadcasted its "fwd"-access eg. DB0AAB-7
Therefore the connect path is in Pseudo-Syntax:

    DB0AAB-7 v OE1XLR < OE1XAB-5

Note: Some software is using only a single SSID for both - user-logins and forwarding, some mix them. Here we are only talking about the SSID used for forwarding.
(FBB,DPBOX: same ssid for user-logins and forwarding, BCM: different ssid, but BBS-forwarding is for historical reasons also accepted on SSID which should only be used for user-logins; DFWD should never use the SSID used for user-logins on BCM).

How to know that BBS B is available on the network?

This is up to the implementation. The BBS might track the Flexnet-Routing information, but it could also just try to connect and see.

What should happen if the mail can not be forwarded?
-Timeout: try DFWD for eg. 6min, if that fails use "normal" forward

The exact timeout (or if DFWD is tried at all) is up to the implementation.

The actual forwarding
is done as usual. If there is more than one mail in the queue, more than one mail is forwarded. If there is mail from B to A it is also forwarded. The only limitation is:
- personal
- to <call>@<local_bbs> (no relaying)
- from <call>

There is no special SID flag used.

Security?
Usually forward is secured by a password, but here only public key algorithms could be used for practical reasons.
But: There is no relaying, no bulletins are forwarded, therefore a "bad" user can not do more than he could do when logging into some BBS using any call.
Still: No WP-broadcasts should be accepted in DFWD, no logins from BBSs which are not known to support DFWD should be accepted.

If there is some password set (eg. some sysops have aggreed on a "common" password), that password and password-method should be used.

32. FAQ - oft gestellte Fragen und Antworten

F: Ist ein Umstieg von FBB/DieBox/MSYS auf OpenBCM möglich?
A: Für FBB und MSYS gibt es keine Utilities zum Konvertieren der Datenbestände. Hier ist es am Besten, die Daten über ein Kabel (KISS oder AXIP) von der "alten" Box auf die "neue" Box zu forwarden, falls das die "alte" Software zulässt. Für DieBox gibt es sowohl eine Software zum Konvertieren der Mails als auch eine Möglichkeit zu Übernahme der Benutzerdaten:

- Mit dem Programm BOX2BCM von OE6BUD können die Mails konvertiert werden. Mails, die einen AutoBIN-Teil enthalten, werden dabei allerdings nicht konvertiert. BOX2BCM generiert aus den Einträgen im INFO-Verzeichnis die neuen Dateien für die OpenBCM-Mailbox. Das Programm BOX2BCM muss dazu im Boot-Verzeichnis stehen, ebenso müssen die Dateien LIFETIME.BOX und CONFIG.BOX der DieBox noch vorhanden sein. Der Pfad für das Info-Verzeichnis wird aus der Datei CONFIG.BOX gelesen. Die Lifetime für einen Eintrag wird aus LIFETIME.BOX gelesen. Achtung: Bei DieBox bedeutet die Lebensdauer "0", dass die Mail nie gelöscht wird, während das bei der OpenBCM-Mailbox bedeutet, dass die Mail beim nächsten Purge gelöscht wird. Für eine unendliche Lebensdauer muss bei der OpenBCM-Mailbox "999" angegeben werden. Da das Programm schon relativ alt ist, kann es zu Fehlern beim Import kommen.
- Die Benutzerdaten aus der DieBox-Datei *user3.idx* können von der OpenBCM-Mailbox direkt importiert werden (Befehl UIMPORT).

F: Bei der DOS-Version wird nach einigen Minuten der Monitor dunkel - ist etwas defekt?
A: Nein, das ist der Bildschirmschoner, er kann mit dem Befehl "crtsave 0" abgeschaltet werden.

F: Bei der Mailbox DB0XYZ läuft schon eine neue Version, woher kann ich diese Version bekommen?
A: Neue Versionen werden in das Board BAYBOX mit dem Verteiler BAYCOM eingespielt. Sollte die neue Version noch nicht dort zu finden sein, so handelt es sich um eine Beta-Version, also um eine Testversion, welche bei einigen wenigen Boxen gestestet wird, bevor sie verteilt wird. Eventuell lohnt sich auch ein Blick nach http://dnx274.dyndns.org/baybox

F: Wenn ein Run-Utility Bulletins löscht, so werden Fernerase-Mails erzeugt. Kann ich etwas dagegen tun?
A: Ja, mit einem Trick: Die Mails nicht löschen, sondern deren Lifetime auf #0 setzen. So werden die Bulletins genauso wie bei ERASE unsichtbar und beim nächsten Purge gelöscht.

F: Warum wird nach dem Hochstarten einer Box mehr freier Speicher beim Befehl VERSION dargestellt, als wenn die Box eine Weile läuft (DOS)?
A: Die Speicherverwaltung der BayCom-Mailbox nutzt die malloc() und free() Aufrufe der C-Laufzeitbibliothek. Diese verwaltet den Speicher jeweils zu Blockgrößen, wie sie angefordert wurde. D.h. wenn zuerst ein sehr großer Block angefordert wird und dann ein sehr kleiner, und dann der große Block wieder freigegeben wird, entsteht eine Lücke. Der angezeigte freie Speicher ist aber stets nur der größte, zusammenhängende Block, und nicht die Summe aller "Schnipsel". Bleibt eine Lücke frei, so kann diese bei der nächsten Anforderung wieder genutzt werden, weil die Verwaltung stets nach dem ersten freien Block sucht, in den der angeforderte Block hineinpasst. Dabei entsteht natürlich ein Verschnitt, weil viele der angeforderten Blöcke eine unterschiedliche Größe haben. So erklärt sich, warum der angezeigte freie Speicher mit der Zeit immer weniger wird: Der Speicher wird immer mehr fragmentiert. Wird ein Block allerdings wieder freigegeben und sind in der Nachbarschaft ebenfalls freie Blöcke, so werden diese wieder zusammengefügt und es entsteht wieder durchgehender, freier Speicher. Der Vorgang der Fragmentierung setzt sich deshalb nicht fort, sondern pegelt sich auf einen durchschnittlichen Wert ein, der dann auf Dauer im Schnitt erhalten bleibt. Manche Blöcke bleiben allerdings sehr lange belegt (z.B. Sprachen, *convert.bcm*, *fwd.bcm* und solche Listen), weshalb auch der freie Speicher lange unverändert zu sein scheint. Üblich ist, dass gegenüber dem maximal möglich verfügbaren Speicher nach dem Hochstarten etwa 30-50 kB abgehen. Dies ist vor allem beim Start von externen DOS-Programmen lästig, da für diese nur durchgehend freier Speicher nutzbar ist.

F: Untersucht man die Datei *bcm.exe*, kommen Befehlswörter und Meldungen zutage, die keine oder eine unbekannte Wirkung haben. Was hat es damit auf sich und warum sind diese nicht dokumentiert?
A: Es gibt in den Quellen zur Box viele Baustellen, an denen irgendeine Funktion angedacht war aber nie realisiert wurde, oder die Teile werden auch zu anderen Zwecken verwendet und erfüllen ihre Funktionalität in der Box nur teilweise. In jedem Fall sollten nur die Funktionen als vorhanden gewertet werden, die auch dokumentiert sind. Da beim Erstellen der Boxsoftware eine relativ penible Versionsverwaltung betrieben wird und alle Erweiterungen automatisch in die Änderungslisten einfließen, gibt es kaum undokumentierte Funktionen, die nicht bewusst undokumentiert geblieben sind.

F: Im Syslog tauchen immer wieder völlig unverständliche Meldungen auf. Wie ist zu unterscheiden, welche Meldungen wirklich wichtig sind und was ignoriert werden kann, ggf. welche Gegenmaßnahmen notwendig werden.
A: Das ist ein sehr wunder Punkt. Zum einen ist klar, dass eine Dokumentation eigentlich dringend nötig wäre, zum anderen gibt es aber so furchtbar viele Meldungen (ca. 400 verschiedne), die potentiell kommen können, dass es ein gewaltiger Aufwand wäre sie alle zu dokumentieren. Prinzipiell kann gesagt werden, dass die Kategorie "#L" meist völlig harmlos ist, "#S" ist je nach Laune des Entwicklers entweder schlimm oder nicht, hingegen "#F" und "#A" sind stets höchst unangenehm und führen grundsätzlich dazu, dass das gewünschte Ergebnis nicht erzielbar ist. Eine Liste oft vorkommender Meldungen wird baldmöglichst nachgereicht.

F: Eingegebene Nachrichten werden sehr schnell an andere Boxen weitergegeben. Dies nimmt die Möglichkeit, nochmals über den Inhalt einer abgeschickten Nachricht nachzudenken und diese ggf. noch vor Weitergabe zu löschen.
A: Die unverzügliche Weitergabe von Nachrichten zu den Nachbarboxen war ein wichtiges Kriterium bei der Entwicklung der OpenBCM-Mailbox. Eine Nachricht sollte stets so schnell wie möglich am Ziel ankommen, sofern dies die Linkstrecken ermöglichen. Es ist deshalb nicht vorgesehen und im momentanen Design auch gar nicht ohne weiteres machbar, Nachrichten verzögert loszusenden. Eine Lösung für Kurzentschlossene ist, vor Abschluss der Nachricht deren Inhalt nochmals zu prüfen und diese ggf. durch CTRL-X oder /AB zu verwerfen und somit gar nicht erst abzuschicken.

F: Unter ALTER REJECT wäre es wünschenswert, mehr Begriffe anzugeben als in die momentan verfügbare 80-Zeichen-Zeile passen.
A: Dieser vielfach geäußerte Wunsch ist angekommen, ist jedoch nur mit sehr viel Ungemach zu verwirklichen. In der derzeitigen User-Struktur sind dafür nur 80 Zeichen vorgesehen und ein Umorganisieren dieser Struktur würde einen enormen Aufwand für alle Boxbetreiber beim Update bedeuten. Es wäre zum einen sehr viel freier Plattenplatz (ca. 100 MB) erforderlich, zum anderen wäre ein Zurückrüsten auf alte Versionen nicht mehr möglich. Außerdem würde das reorganisieren der Daten vor dem Update ca. 1 Stunde die Box lahm legen. Bei der Vielzahl der laufenden Boxen lohnt es sich durchaus, für einen solchen Schritt mehrere Änderungen auf einmal zusammenzufassen und erst dann eine solche Umorganisation durchzuführen.

F: Es war einmal eine 32 Bit-DOS-Version der OpenBCM-Mailbox angedacht und wurde auch angekündigt. Was ist daraus geworden?
A: Es gab eine 32 Bit-DPMI-Version von *bcm.exe*. Diese hat sehr viel Mühe bei der Portierung bereitet, da viele unter DOS mögliche Zugriffe im protected mode ganz anders sind, erst recht wenn diese von einem 32 Bit-Programm aus erfolgen. Auf den ersten Blick sieht dann alles auch ganz gut aus, man kann frei Speicher adressieren, es existiert eine virtuelle Speicherverwaltung, die sehr großzügige Speicherzugriffe ermöglicht. Außerdem entfällt die modulweise Segmentierung, was den Code angenehm übersichtlich macht. Es hat sich jedoch gezeigt, dass auch ganz krasse Nachteile entstehen, die das Projekt letztlich dann nach dessen Fertigstellung zu Fall gebracht haben:

- Zugriffe auf AX.25-Treiber (in diesem Fall *l2.exe*) sind extrem langsam, da bei jedem einzelnen Zugriff sehr viel im DPMI-Kern abläuft.
- Auch DOS-Zugriffe jeglicher Art sind ausgesprochen langsam, aus gleichem Grund.
- Hardware-Interrupts werden sehr häufig verschluckt, da auch diese vom DPMI-Kern geregelt werden müssen und während Umschaltvorgängen oft für lange Zeit gesperrt sind. Dadurch verliert *l2.exe* sehr häufig Pakete.
- Eine Zeitsynchronisierung mittels Timer-Interrupt ist sehr schwierig und anfällig, insbesondere weil es nicht gelingt, die Serviceroutine vom Memory-Paging auszuschließen und dadurch die Stabilität in Frage steht.
- Ansteuerung der Hardware ist nicht ohne weiteres möglich, dadurch keine serielle Schnittstelle und kein Watchdog etc. Auch der Bildschirm ist sehr langsam, weil intern aufwendiger.
- Insgesamt läuft die Box im 32 Bit-Mode um etwa einen Faktor 2 langsamer.
- Die DPMI-Bibliothek von Borland ist als Speicherfresser zu betrachten. *bcm32.exe* läuft nicht ohne weiteres auf einem mit 4 MB (!) bestückten Rechner, zumindest bleibt dann keinerlei Platz mehr für Dateisystem-Puffer. Das unterste MB wird faktisch hergeschenkt.

Obige Argumente machen klar, warum das Projekt nach näheren Tests zu den Akten gelegt wurde. Schade um die Arbeit und die dadurch entstandenen Kosten. Speziell bei letzterem ist ein besonderer Dank an DL6FBS zu richten, der hier den wesentlichen Teil übernommen hat.
In der Zwischenzeit gibt jedoch es eine Version der OpenBCM-Mailbox für Windows.

F: In der PS-Liste stehen oft nicht nur eingegebene Kommandos, sondern auch Symbole, die offenbar vom Programm generiert werden. Was bedeuten diese?
A: Beim Forwarding werden hier normalerweise die übertragenen Mails bzw. die ausgetauschten Kommandos dargestellt. Wird gerade keine Mail gesendet oder empfangen, so ist im Kommandoteil der PS-Liste der Zustand der Forward-Routine zu entnehmen. Das bedeutet im Einzelnen:

fwd_loop
Die Forward-Hauptschleife wurde aufgerufen

fwd_send
Die Forward-Schleife verzweigt demnächst zu einem SEND-Befehl

fwd_delay
Es wurden Nachrichten gesendet und es wird abgewartet, ob noch eine Mail für den Partner eintrifft bevor die Verbindung beendet wird

fwd_rec
Die Forward-Maschine erwartet ein Kommando vom Partner

fwd_pro
Es wurde ein Prompt gesendet

fwd_connect
Ein Connect-Befehl wurde abgesetzt, Verbindung wird erwartet

fwd_connected
Die AX.25-Verbindung ist gelungen

wait_for_sid
Ein Forward-Partner wurde connected, und es wird nun die SID des Partners erwartet

wait_pr1
Es wird der erste Prompt nach dem Login erwartet

wait_pr2
Es wird der zweite Prompt nach dem Senden der SID erwartet

rx_wait_sid
Ein Forward-Connect wurde entgegengenommen und die eigene SID gesendet; es wird nun die SID des Partners erwartet

F: Was bedeutet "NO - BID" beim Forward?
A: Das ist kein Fehler (also kein BID-Mangel o. ä.), sondern zeigt an, dass die angebotene Bulletin bereits lokal vorhanden ist und deshalb vom Partner nicht geschickt werden soll.

F: Darf die Baycom-Mailbox im CB-Funk benutzt werden?
A: Die OpenBCM-Mailbox ist für den CB-Funk freigegeben. Es handelt sich hierbei um eine mündliche Vereinbarung mit dem Autor, um einen Ansturm an Support-Fragen zu vermeiden. Fragen aus dem CB-Bereich zu Einstellungen, Installation, Bugreports, Programmierung und Sonstigem deshalb bitte an Jonas, DO1MJJ/DJJ812 oder per Email an "djj812@dbo812.de"/ PR-Mail: "DJJ812@DBO812.#M.OBB.BAY.DEU.EU" richten. Ebenso ist die Internetseite http://www.dbo812.de/ die offizielle Homepage für die CB-Version der OpenBCM-Mailbox. Es müssen ebenso die Rufzeichenbedingungen der RegTP, der Regulierungsbehörde für Telekommunikation und Post, erfüllt sein.

F: Wie kann ich die OpenBCM-Mailbox für CB-Funk umstellen?
A: Der Rufzeichensatz für CB-Rufzeichen wird mit dem Befehl CALLFORMAT und evtl. auch noch LOGINCALLFORMAT aktiviert, wobei ersterer für nur für den Forward und der zweite für die Userlogins gilt. Dieser muss vor dem 1. Start der OpenBCM-Mailbox am besten manuell in die *init.bcm* eingetragen werden. Die Position spielt dabei keine Rolle. Als Parameter wird eine Ziffer verwendet:

0 = Amateurfunkrufzeichen
1 = CB-Rufzeichen
2 = Beide Rufzeichengruppen

Für CB-Funk muss also "CALLFORMAT 1" oder "CALLFORMAT 2" eingetragen werden. Ist CALLFORMAT auf 1 oder 2 gesetzt, wird auch "(CB)" als Hinweis für Aktivierung der CB-Version hinter jeder Mailbox-Versionsangabe mit angezeigt.

F: Welche Rufzeichen darf ich im CB-Funk verwenden?
A: Es besteht im CB-Funk zwar keine Rufzeichenpflicht, allerdings dürfen keine bereits vergebenen Rufzeichen, sei es angemeldete CB-Rufzeichen, Amateurfunkrufzeichen oder Rufzeichen irgendeines anderen Funkdienstes, verwendet werden! Generell wird die Anmeldung einer Mailbox auf CB empfohlen, um eine gewisse Organisation zu ermöglichen. Mehr hierzu auf der Homepage der RegTP, der Regulierungsbehörde für Telekommunikation und Post, unter http://www.regtp.de.

F: Beim Versuch in den Filesurf zu kommen, bekomme ich immer die Meldung "filesurf not enabled".
A: Der Filesurf ist nur fester Bestandteil der Linux- und Windows-Version. Er muss erst mit dem Sysopbefehl "fspath" aktiviert werden, wobei der angegebene Pfad vorhanden und für die Userrechte der OpenBCM freigeschaltet sein muss. Für DOS ist ein externes Programm wie z.B. EL besser, da hier der Arbeitsspeicher so knapp ist, dass die Filesurf-Option meist nicht mit einkompiliert wird.

F: Warum funktioniert der eingebaute Texteditor "edit", der Rejecteditor "REJEDIT", der Convert-Editor "CONVEDIT" oder der Forward-Editor "FWDEDIT" nicht?
A: Die eingebauten Editoren sind in der DOS-Version häufig nicht mit in die Mailbox einkompiliert, da sie einiges von dem raren Hauptspeicher fressen. Unter Linux/Windows sind diese Editoren jedoch normalerweise immer verfügbar.

F: Kann ein User die Packetlänge für sich einstellen?
A: Der Sysop kann für jeden User mit Hilfe des SETUSER-Befehls die Paketlänge (PACLEN) einstellen, dies gilt auch für die Forward-Partner. Der Eintrag in der "ALTER"-Liste erscheint nur, wenn die Packetlänge größer als 0 ist.
Die globale Paketlänge kann mit "PACLEN" eingestellt werden. Der Wert darf zwischen 40 und 255 liegen, bei dem Wert 0 wird die maximale Länge verwendet.

F: Wie aktiviere ich den Autorouter?
A: Der Befehl dazu heißt "autofwdtime". Er bestimmt, wie alt eine eingetroffene Nachricht in Tagen sein darf, um aus ihr den Pfad auslesen zu dürfen. Ist der Wert 0, ist der Autorouter deaktiviert. Zusätzlich muss der Befehl "afwdlist" in die Datei *crontab.bcm* eingetragen werden und dort mindestens einmal täglich durchlaufen werden.

F: Wie hebe ich das Download-Limit auf?
A: Der Befehl dazu heißt "setuser status 1". Die Größe des Limits kann mit dem Befehl "userquota" in KB angegeben werden.

F: Arbeitet die BCM32 mit Flex32 zusammen?
A: Ja! Dazu muss in der *init.l2* nur folgender Eintrag gemacht werden:
```
assign axip
peer 127.0.0.1
port 4722
txport 4721
```
Bei Flex32 muss im Control-Center zusätzlich ein AXIP-UDP-Port erstellt werden, der dann mit Port 4721 empfängt und mit Port 4722 sendet.

F: Arbeitet die BCM32 mit XNET(NT) zusammen?
A: Ja! Dazu muss in der *init.l2* nur folgender Eintrag gemacht werden:
```
assign axip
peer 127.0.0.1
port 4724
txport 4723
```
Bei XNET muss in der *autoexec.net* ein AXIP-UDP-Port hinzugefügt werden, der dann mit Port 4723 empfängt und mit Port 4724 sendet. Dies ist beispielsweise ab Version 1.30 von XNET mit folgender Zeile machbar:
```
att ip0 axudp 0 1 l4723 d4724 127.0.0.1
```

F: Wie bekomme ich die OpenBCM unter Windows9x ans laufen?

A: Unter Windows9x (95, 98, Millenium) kann nur die DOS-Version der OpenBCM-Mailbox verwendet werden. Diese läuft bisher nur mit Flexnet für DOS oder Flex95 zusammen. Hier muss allerdings versucht werden, konventionellen Speicher, also unterhalb der 640KB-Grenze, freizubekommen. Ein Betrieb mit Flex32 oder dem Flexnet DOS-Node ist unter Windows 9x nicht möglich.

F: Läuft die DOS-OpenBCM unter Windows9x mit XNET zusammen?
A: Die OpenBCM für DOS benötigt Flexnet als Treiber. Es ist möglich, Flex95 (übrigens auch Flex32) mit XNET(NT) zu verbinden. Die funktioniert mit AXIP-UDP-Ports. In XNET(NT) kommt folgender Eintrag in die *autoexec.net*:

```
att ip0 axudp 0 1 l4726 d93 127.0.0.1
```

Hierzu muss Flex95 VOR Windows mit folgenden Parametern gestartet werden:

```
flexnet
ether32 /c=1 /n0=AXIP_UDP,127.0.0.1:4726
flex
tfemu (nur notwendig wenn auch ein DOS-Terminal mitlaufen soll)
```

Bei Flex95 müssen die Einstellungen im Control-Center vorgenommen werden. Flex32 wird von der DOS-OpenBCM nicht unterstützt.
Achtung: Werden weitere AXIP-UDP-Ports in XNET benutzt, dürfen diese nicht auf dem UDP-Port 93 empfangen, da Flex95 diesen bereits belegt und nicht geändert werden kann!

F: Läuft die OpenBCM für DOS mit Flex32 unter Windows?
A: Ein Betrieb mit Flex32 oder dem Flexnet DOS-Node ist unter Windows 9x nicht möglich. Nur der Betrieb der OpenBCM für Windows ist mit Flex32 oder XNET möglich.

F: Wie unterdrücke ich den Forward von 7PLUS und AUTOBIN-Dateien?
A: Einfach beim jeweiligen Forward-Partner in der *fwd.bcm* die Option "-K", "-L", "-M" oder "-N" angeben.

F: Die Zeitzone wird in der OpenBCM nicht richtig angezeigt, warum arbeitet die OpenBCM überhaupt mit unterschiedlichen Zeiteinstellungen?!
A: Mit dem Befehl "time -a" lässt sich die Systemzeit und die Differenz zur UTC (Welt)-Zeit anzeigen. Diese Einstellung muss unbedingt stimmen, damit u. a. der Autorouter richtig funktioniert.
Unter Windows kam es hier vermehrt in alten Versionen zu Problemen, insbesondere dann, wenn unter der Systemsteuerung unter "System", "Erweitert" eine Umgebungsvariable "TZ" falsch definiert wurde. Am besten ist, man löscht diese Umgebungsvariable ganz und verlässt sich nur auf die Systemeinstellungen des Betriebsystems (die natürlich richtig eingestellt sein sollten). Falls man die Variable unbedingt nutzen muss, weil z.B. ein anderes Programm diese Variable zwingend benötigt, sollte sie in Westeuropa möglichst auf "MET+1MEZT+2" eingestellt sein. Die OpenBCM verfügt über eine automatische Sommer-/Winterzeit-Umstellung, sollte dies nicht gewünscht sein, so ist die Variable "summertime" in *init.bcm* auf "0" zu setzen, ansonsten sollte die Variable auf "1" stehen.

F: Das Telefonmodem reagiert nicht in der DOS-Version.
A: Der IRQ muss mit der Schnittstelle übereinstimmen. Das Echo des Modems muss in einem normalen Terminalprogramm unter der selben Schnittstellengeschwindigkeit wie in der OpenBCM mit ATE0 ausgeschaltet werden und die Einstellung mit ATW gespeichert werden.

F: Die OpenBCM beendet sich ständig selbst.
A: Das liegt vermutlich an zu alten MSG-Dateien. Ein Hinweis darauf ist die Meldung "missing lines" in der *trace/syslog.bcm*. Das Format der Dateien hat sich ab der BCM v1.43 geändert und alte Sprachdateien sind immer noch im Umlauf. Durch das Austauschen mit aktuellen MSG-Dateien sollte das Problem behoben sein. In den aktuellen Versionen kann man mit "a s" auch die Versionsnummer der MSG-Dateien abfragen. Sollte hier keine Versionsnummer stehen, sind die MSG-Files definitiv veraltet!

F: Wieso verlangt die Linux/Windows-OpenBCM ein Passwort bei Login via Telnet/HTTP?

A: Jeder Benutzer benötigt ein eigenes Passwort, um sich via HTTP oder Telnet einloggen zu können. Dies stellt der Sysop mit dem Befehl "setuser ttypwd" ein. Es darf max. 8 Zeichen lang sein.

F: Änderungen an der Konfiguration werden nicht erkannt.
A: Damit die Änderungen erkannt werden, muss NEW eingegeben oder die OpenBCM neu gestartet werden. Beim Speichern der Konfigurationsdateien muss auf die Endung *.bcm* geachtet werden. Bis auf die *init.l2* haben alle Dateien diese Endung. Im Zweifelsfall verwendet einen anständigen Texteditor (unter Windows NICHT Wordpad - empfehlenswert ist z.B. Textpad, siehe http://www.textpad.com) der auch CR/LF am Zeilenende beherrscht.

F: Woher kommen Übertragunsfehler / CRC-Fehler?
A: Meistens treten diese auf einem KISS-Link auf. Um dies zu verhindern muss der CRC-Modus aktiviert werden. Dies geschieht bei Flexnet mit der Option "c" hinter der Baudrate mit "fset" und bei XNET durch Angabe der Port-Mode-Parameters "2".

F: Wie schließe ich ein TNC2 an die OpenBCM an?
A: Ein TNC kann von der OpenBCM nur mit KISS direkt angesteuert werden, es ist jedoch davon ABZURATEN! Besser ist die Verwendung einer geeigneten Netzknotensoftware zwischen Mailbox und Außenwelt.

F: XNET(NT) und AXIP-Links (auch dynamische IPs) – wie geht das?
A: Bei festen IP-Adressen reicht ein Eintrag in der *autoexec.net*:

```
att ip0 axudp 0 1 l93 d93 192.168.1.1
```

"ip0" sowie "0" sind die jeweiligen IP- und XNET-Ports, l93 und d93 der Empfangs- und Sende-Port für AXIP-UDP-Packete und 192.168.1.1 die IP-Adresse des Linkpartners. Die angegebenen UDP-Ports müssen in einer evtl. vorhandenen Firewall freigeschalten sein. Möchte man über das Internet eine Node-Verbindung aufbauen, ist es in den meisten Fällen nicht möglich, da die Nodes über dynamische IP-Adressen verfügen. Damit diese Programme arbeiten können, benötigt man einen DNS-Eintrag, den man z. B. bei www.dyndns.org oder www.myip.org bekommen kann. Um den Abgleich der IP-Adressen zu diesen DNS-Einträgen zu automatisieren, gibt es wieder entsprechende Tools (DynDNS- Clients). Unter Windows hat sich das Programm "ynSite" dafür sehr bewährt. Um die IP-Adressen auch im entsprechenden AXIP-UDP-Treiber im XNET(NT) einzustellen, bedarf es eines weiteren Tools, das zyklisch gestartet werden muß und anhand der DNS-Einträge die neuen IPs einstellt, z.B. NTNETDNS (auf der Downloadseite von http://www.dbo812.de zu finden. Im Archiv befindet sich auch eine Installationsanleitung dazu.

F: Flex95 und AXIP-Links (auch dynamische IPs) – wie geht das?
A: Notwendig sind dafür nur die entsprechenden Treiber in *autoexec.bat* (bevor Windows gestartet wird). Der AXIP-Port wird unter DOS mit dem Treiber IPPD realisiert. Hier sind keine dynamischen IPs möglich und auch nur LAN-interne AXIP-Links. Unter Windows geht es mit dem Treiber ETHER32, allerdings sieht es damit besser aus.

- Unter DOS:
  Für IPPD wird vor dem Laden von Flexnet noch ein sog. Packet-Treiber für die Netzwerkkarte benötigt, der meistens im Lieferumfang bei den Treiberdisketten dabei ist. Hierbei handelt es sich meistens um eine kleine, speicherresidente *.com*-Datei, der als Parameter meistens die IO-Adresse, der Hardware-IRQ und der Software-IRQ, den man sich aussuchen kann, der aber meistens 0x60 ist, angegeben werden muss. Unter Windows muss das TCP/IP-Protokoll installiert sein. Folgende Batchdatei kann verwendet werden:

  ```
  ne2000 0x300 10 0x60
  flexnet
  ippd -c:1 -m:192.168.1.1 -p:192.168.1.2 -u:93 -i0x60
  flex
  fset digicall MYCALL
  tfemu
  ```

192.168.1.1 ist im Beispiel die IP-Adresse des Rechners auf dem die Batchdatei gestartet wird, 192.168.1.2 ist die des Nachbarrechners. 93 ist der UDP-Port und 0x60 die Nummer des Software-Interrupt des verwendeten Packet-Treibers. Die Angabe von c=1 bestimmt die Anzahl der AXIP-Ports in Flexnet, alle weiteren Ports können mit dem Zusatzprogramm IPPDCFG eingestellt werden.

- Unter Windows:
Für ETHER32 braucht man weitaus weniger Angaben, da die eigentlichen Netzwerkaufgaben Windows übernimmt. Folgende Batchdatei kann verwendet werden:

```
flexnet
ether32 /c=1 /n0:AXIP_UDP,192.168.1.2
flex
fset digicall MYCALL
tfemu
```

192.168.1.1 ist im Beispiel die IP-Adresse des Rechners auf dem die Batchdatei gestartet wird, 192.168.1.2 ist die des Nachbarrechners. 93 ist der UDP-Port. Bei ETHER32 ist nur die Angabe der Nachbar-IP notwendig, da er die Netzeinstellungen von Windows benutzt. Die Angabe von c=1 bestimmt die Anzahl der AXIP-Ports in Flexnet, alle weiteren Ports können mit dem Zusatzprogramm ETH32CFG eingestellt werden. Will man nun über das Digicall linken und man hat mehrere Ports installiert, muss man als SSID nur die Portnummer angeben. z. B. MYCALL-1 um über Port 1 zu linken.

Bei dynamischen IP-Adressen empfiehlt sich die Verwendung von DNS-Einträgen, hierzu kann man unter Windows/Flex95 das Programm ETH32DNS benutzen.

F: Ständig kommt beim Senden von Nachrichten "Befehl oder Dateiname nicht gefunden".
A: In der DOS-Version 1.43 bis 1.44 der BCM war dies ein Fehler der BCM und trat meist nur auf, wenn sehr wenig Arbeitspeicher vorhanden war. Nachfolgende Versionen, so auch die OpenBCM, haben diesen Fehler nicht mehr. Es wird also Zeit für ein Update!

F: Wie lege ich Forward-Zeiten in der OpenBCM fest?
A: Jeder Eintrag in der *fwd.bcm* beginnt mit der Kopfzeile, die das Boxcall, die Zeitzeile und den Connect-Pfad enthält, z.B.:

```
CB0BOX AAAAAAAAAAAAAAAAAAAAAAAP CB0NOD / CB0BOX CB0DIG
 THEBOX $WP
```

Die Zeitzeile besteht dabei aus den Stunden des Tages, jeweils ein Buchstabe für eine Stunde beginnend mit 0 Uhr bis 23 Uhr. Soll also z.B. um 1 Uhr Nachts der Forward gestartet werden muss der zweite Buchstabe ein "A" sein.
A = Forward von Info & Usermails
U = Forward von Usermails
P = Postforward
. = kein Forward
Weitere Informationen zur Forwarddatei sind im Kapitel "9. Forwarding komplett einrichten" ausführlich beschrieben!

F: Beim Start der OpenBCM kommt die Meldung "Windows 95 or Win32s is not recommended for bcm32!"
A: Die Windows-Version der OpenBCM ist nur für Windows-Versionen mit NT-Kern geeignet. Also nur Windows NT 4.0, Windows 2000 und Windows XP (und folgende). Windows 95, 98, und ME sind nicht geeignet. Dort kann nur die DOS-Version mit Flex95 als Interface benutzt werden.

F: Bei Verwendung der Windows-Version der OpenBCM unter Windows XP kann ich keine Mails speichern, im Trace sieht man die Fehlermeldung:
"#F savemail: fopen […] errno=2 No such file or directory" - ist das ein Fehler der Software, denn unter Windows NT oder 2000 tritt dies nicht auf?

A: Unter Windows XP ist es wohl nötig die Pfade in der Datei *init.bcm* komplett anzugeben, also z.B. "userpath c:/bcm/user/". Unter Windows NT/2000 reicht die Angabe von "userpath user".

F: Ich sehe im Syslog immer nur z.B. folgende Ausgabe "#L savemail: m_filter '/usr/bin/m_filter.prg' not found!" - woran liegt das?
A: Ist das M_Filter <Programm> nicht vorhanden, oder ist der Pfad falsch eingestellt, kommt es beim Senden von Mails zu einer solchen Ausgabe im Syslog. Ferner könnte unter Linux auch noch die Möglichkeit bestehen, dass die Dateizugriffsberechtigungen des M_Filter-Programmes falsch gesetzt sind (es muss mindestens 'rx' gesetzt sein). Um das Mailfilterprogramm ganz zu deaktivieren, muss nur die Option M_FILTER OFF in der Datei *init.bcm* gesetzt werden.

F: Ich sehe im Taskmanager immer einen Connect der eigenen Mailbox und die Auslastung des Systems steht dann die ganze Zeit auf 100%. Die Festplatte rappelt. Der Zustand dauert scheinbar ewig. Ist die Mailbox abgeschmiert?
A: Vermutlich steht z.B. im Connecttext das "checkcount"-Makro. Connectet dann z.B. ein Forward-Partner und hat der noch nie einen Mail"Check" im System ausgeführt, so zählt das "checkcount"-Makro alle "neuen" Mails zusammen. Bei einer größeren Mailbox mit vielen Rubrikmails kann das dann schon mal länger dauern... als Abstellmaßnahme sollte man also das "checkcount"-Makro nicht im Connecttext der Mailbox oder bei "A C" global verwenden!

F: Kann man Mails automatisch in eine Forward-Datei exportieren?
A: Klar geht das - dazu ist nur eine Zeile in *crontab.bcm* mit dem Befehl einzufügen, z.B.

```
fwdexp db0xyz fwd/export/db0xyz.exp
```

Sind für mehrere Mailboxen Forwarddateien zu erstellen, so ist es im Allgemeinen übersichtlicher, eine Datei *mailexport.imp* mit den jeweiligen Export-Zeilen anzulegen und dann nur diese eine Datei über *crontab.bcm* mit z.B.

```
20 6,18 * * * mailexport
```

aufzurufen.
Es ist ferner zu beachten, dass die gemailte Forward-Datei anschließend gelöscht wird, bevor die Mailbox einen erneuten Mailexport-Aufruf macht - sonst bekommt der Forwardpartner die gleichen Mails mehrfach!

F: Wenn man um 15:00 Uhr eine Mail in der OpenBCM verfasst, und diese Mail dann wieder ausliest, wird die Uhrzeit 13:00 Uhr anstatt 15:00 Uhr angezeigt, das ist doch nicht richtig oder?
A: Doch, das ist alles korrekt! In allen Mailboxsystemen wird mit Weltzeit (auch UTC oder GMT Zeitzone genannt) gearbeitet. Die deutsche Sommerzeit unterscheidet sich hiervon um plus 2 Stunden, die deutsche Winterzeit um plus 1 Stunde. Wenn man den Befehl "time -a" in der OpenBCM eingibt, bekommt man sowohl die lokal eingestellte Uhrzeit, als auch die Weltzeit angezeigt.

F: Ich bekomme den automatischen HTTP Zugang für Gäste einfach nicht zum Laufen. Habe folgende Einstellungen:

```
HTTPACCOUNT 1
HTTPGUESTFIRST 1
GUESTCALL GAST
```

Sobald aber jemand die Mailbox per HTTP "connected", wird er immer noch nach dem Benutzernamen und Passwort gefragt. An was könnte das liegen?
A: Für den User "GAST" ist vermutlich kein TTYPW gesetzt. Um dies zu überprüfen, einfach mal als Sysop "A GAST" abfragen und den Zahlenwert bei TTYPWlen überprüfen. Ist er auf "0", so ist kein Passwort eingestellt. Um dann ein Passwort zu setzen, reicht ein "setuser gast ttypw xyz123" aus. Da HTTPGUESTFIRST auf 1 gesetzt wurde, wird dann der GAST-User jedes Mal automatisch ohne Passwortabfrage eingeloggt.

F: Ich habe einen Forward von meiner Mailbox DB0ABC-3 zu einer anderen Mailbox DB0XYZ-8 meines Erachtens richtig eingerichtet, jedoch kommt ein Mailaustausch immer noch nicht zustande! Im Trace-Fenster sehe ich, dass die andere Mailbox DB0XYZ-8 nach einem Verbindungsausbau immer "*** no logintime" sendet und dann die Verbindung wieder kappt:

```
12:fm DB0ABC-3 to DB0XYZ-8 ctl I00^ pid F0 [29]
[OpenBCM-1.03-AB1D1FHMRW$]
>
12:fm DB0XYZ-8 to DB0ABC-3 ctl RR1v
12:fm DB0XYZ-8 to DB0ABC-3 ctl I10^ pid F0 [255]
*** no logintime
```

Bei einem Connectversuch von DB0ABC-3 nach DB0XYZ-8 passiert etwas ähnliches:

```
12:fm DB0XYZ-8 to DB0ABC-3 ctl I15^ pid F0 [40]
[OpenBCM-1.03-AB1D1FHMRW$] 1406031330
>
12:fm DB0ABC-3 to DB0XYZ-8 ctl RR6v
12:fm DB0ABC-3 to DB0XYZ-8 ctl I61^ pid F0 [25]
*** no password expected
```

Woran liegt das, und wie kann ich den Forward richtig einrichten?
A: Bei der Partnermailbox DB0XYZ-8 (ebenfalls ein OpenBCM-System) ist offensichtlich ein Diebox-Passwort für den Forward eingerichtet (Datei *db0abc.pwd*). Vielleicht war die eigene Mailbox DB0ABC-3 vorher schon mal unter einem Diebox-System ein Linkpartner von DB0XYZ-8. Wenn man bei der Mailbox DB0XYZ-8 die Datei *db0abc.pwd* löscht oder verschiebt, funktioniert der Forward, allerdings ist dieser dann nicht mehr passwortgeschützt. Alternativ könnte man auch bei der Mailbox DB0ABC-3 die gleiche Datei, aber unter dem Namen *db0abc.pwd* anlegen. Noch besser wäre es jedoch, bei beiden Systemen auf die PWD-Dateien zu verzichten und für den Forward das MD5-Passwortverfahren anzuwenden, d.h. in beiden Mailboxen für das jeweils andere Mailboxrufzeichen mit "SETUSER <call> SF MD5" das MD5 Verfahren aktivieren, um dann mit "SETUSER <call> PW <passwort>" das entsprechende (gleiche) Passwort zu setzen.

F: Bei STATUS FORWARD wird ja eine Statistik zum Forward angezeigt. Offensichtlich hat der Autor bei den Statusmeldungen einen Tippfehler eingebaut, ab und an erscheint dort "recv" und mal "recvp" bzw. "send" und mal "sendp".
A: Nein! Das ist alles korrekt! Bei "recv" wird tatsächlich gerade eine Mail empfangen, bei "recvp" findet ein receive-proposal (=Empfangsvorschlag) mit der anderen Mailbox statt. Analog gilt dies auch für "send" und "sendp". Alle Statusmeldungen sind übrigens bei HELP STATUS FORWARD genau erklärt.

F: Beim Forward zwischen DIEBOX und OpenBCM werden bei Usermails die MIDs nicht übergeben, woran liegt das?
A: DIEBOX ab v1.9c1 kann auch MIDs verwalten, daher funktioniert dies auch prinzipiell. Bei DIEBOX gibt es aber eine Datei *mbsys\sfwid.box* in der für BCM und OpenBCM folgende Einträge zu machen sind:

```
BayCom-1.1  18 S
BayCom-1.2  18 S
BayCom-     19
OpenBCM-    19
```

Fehlt die letzte Zeile, dann passiert das in der Frage beschriebene Verhalten.

F: Ich will einen Forward-Connect via Telnet machen. Es geht aber kein Connect raus, was mache ich bloß falsch?
A: Evtl. ist in der *fwd.bcm* die Zeile für den Connect-Pfad falsch eingestellt. Ein üblicher Fehler ist etwa die Zeile mit

```
DBO123 AAAAAAAAAAAAAAAAAAAAAAAP telnet://dbo123.dyndns.org:4719
```

anstelle von

```
DBO123 AAAAAAAAAAAAAAAAAAAAAAAP telnet:dbo123.dyndns.org:4719
```

abzugeben (das :// ist durch ein : zu ersetzen!).

F: Ich nutze OpenBCM als User-Mailbox. Wenn ich eine Mail eingebe wird diese auch wie gewünscht sofort zur Heimatmailbox gesendet. Neue Mails in der Heimatmailbox werden jedoch nur dann abgeholt, wenn ich selber eine Mail dorthin schicke. Normalerweise sollte die doch durch den Forward-Eintrag in *crontab.bcm*

```
...
3,33      *    *    *     *       forward
....
```

halbstündlich erfolgen, zumal im SLOG folgender Hinweis kommt:

```
startfwd: checking DB0CZ
```

Was geht hier bloß schief?
A: Vermutlich ist in der Forwarddatei *fwd.bcm* folgender Eintrag zu finden:

```
DB0CZ AAAAAAAAAAAAAAAAAAAAAAAA DG1GGG / DB0CZ
```

Damit der Forward auch tatsächlich nach DB0CZ angestoßen wird, ist statt "A" ein "P" einzugeben, also

```
DB0CZ PPPPPPPPPPPPPPPPPPPPPPP DG1GGG / DB0CZ
```

Der Eintrag im slog sagt bei "A" nur aus, dass überprüft wird, ob Mails für DB0CZ vorliegen, nicht aber, dass der Forward auf alle Fälle aufgebaut wird. Dies wird erst mit "P" gemacht.

F: In meinem SYSLOG sehe ich beim Senden einer Mail massenweise Zeilen, wie z.B.:

```
14.12.03 15:44:50l DGT274: #L parse_rej: invalid rej-action:
14.12.03 15:44:50l DGT274: #L parse_rej: invalid rej-action:
14.12.03 15:44:50l DGT274: #L parse_rej: invalid rej-action:
14.12.03 15:44:50l DGT274: #L parse_rej: invalid rej-action:
```

Die Datei *reject.bcm* habe ich überprüft, die Einträge sind aber alle richtig. Woran liegt das?
A: Es liegt an der *reject.bcm*! Dort sind mit Sicherheit Zeilen enthalten, die mit einem Leerzeichen beginnen, und das ist nicht erlaubt! Sollen Kommentarzeilen eingefügt werden, so müssen diese mit einem Semikolon beginnen.

33. Urheberrecht

Für die BayCom-Mailbox gilt die "GNU General Public License" Version 2.0, June 1991.

34. Literaturverzeichnis

[1] Sysop-Documentation FBB Version 7.00d, Jean Paul Roubelat, F6FBB@F6FBB.FMLR.FRA.EU, 1997

[2] Sysop-Manual THEBOX Version 1.9, Reinhard Rüdiger, DF3AV@DK0MAV.#NDS.DEU.EU, 1992

[3] Dokumentation zum BayCom-Node 1.54/1.55a, Florian Radlherr, DL8MBT@DB0AAB.#BAY.DEU.EU, 1994, 1995

[4] Manual BayCom-Terminal 1.50, Florian Radlherr, DL8MBT, 1992

[5] Manual BayCom-Terminal 1.60, Florian Radlherr, DL8MBT, 1994

[6] Sysop-Manual MSYS Version 1.19, Mike Pechura, WA8BXN, September 1995

[7] Manuals zur W0RLI-BBS Version 17.5, Henry N. Oredson, W0RLI@W0RLI.OR.USA.NOAM, 1994

[8] Update-Infos THEBOX Version 1.9b4-1.9b5, Reinhard Rüdiger, DF3AV, Juni 1994, Sommer 1995

[9] FlexNet 3.3g/h - Dokumentation, Gunter Jost DK7WJ@DB0ZDF, Juni 1995, 1996

[10] Digipoint 5.07 Dokumentation, Joachim Schurig, DL8HBS@DB0GR.#BLN.DEU.EU, 1997

[11] PC/FlexNet 3.3e - Dokumentationen zu den div. Programmteilen Gunter Jost, Januar 1996

[12] ISO 3166 2 und 3 stellige Ländercodes. 15.12.1993, ISO 3166:1993 (E/F) Maintanance Agency beim DIN Berlin, Tel. +49 30 26 28791-2861

[13] BBS Hierarchical Addressing Protocol x.3.4, TAPR Special Interest Group, Dave Wolf WO5H, März 1995

[14] WinGT für Amateurfunk, Gerd Mitländer, DG8NDL@DB0RT.#BAY.DEU.DEU, Version 1.60, April 1996

[15] Computer Networks, Andrew S. Tanenbaum, 3. Auflage, Amsterdam 1996

[16] RFC821 Internet-Übertragungsprotokoll für E-Mails

[17] RFC822 Internet-Nachrichtenformat für E-Mails

[18] RFC1866 Hypertext Markup Language 2.0

[19] RFC2068 Hypertext Transfer Protocol 1.1